IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicolás Valencia S.
12 April, 2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Methodologies used:
  - Data Wrangling from SpaceX REST API and Web Scrapping.
  - EDA:
    - ❖ Main Charts, Map Visualization and Dashboard creation.
    - ❖ SQL queries.
  - Classification Algorithms.
- Main Findings:
  - The more launches, the better at successful landing outcomes SpaceX becomes.
  - Launch Site matters when predicting the landing outcome.
  - Decision Tree is the best performing classification algorithm, but the most consistent results come from Support Vector Machine.

Introduction

- SpaceY is the brand-new rocket company that wants to enter in the industry in order to directly compete with SpaceX.
- In order to predict the future profits of SpaceX via approximating its functioning costs over making launches of their rockets, we must determine the probability of success for reusing the so called “First Stage” of their rockets.
- As we don’t have the private data of future mission specifications, we’re going to work with public data in order to build a Machine Learning Model for predicting those outcomes.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was gathered from two main sources:
    - ➢ SpaceX REST API.
    - ➢ Web-Scraping SpaceX ‘Falcon 9’ Wikipedia Web Page.
- Dara wrangling process:
  - Several Features were standardized and simplified.
  - One-Hot encoding was used to prepare the data for modeling.

Methodology

Executive Summary

- EDA was performed in various stages:
  - SQL
  - Visualization (Maps and Dashboard)
- Various Classification Models were tested:
  - We looked at which ones were the classification models are mostly used and performed a scoring process for selecting the one with the best performing hyperparameters.
  - Three evaluation metrics were used in this process (Accuracy Score, Jaccard Index and F1 Index).

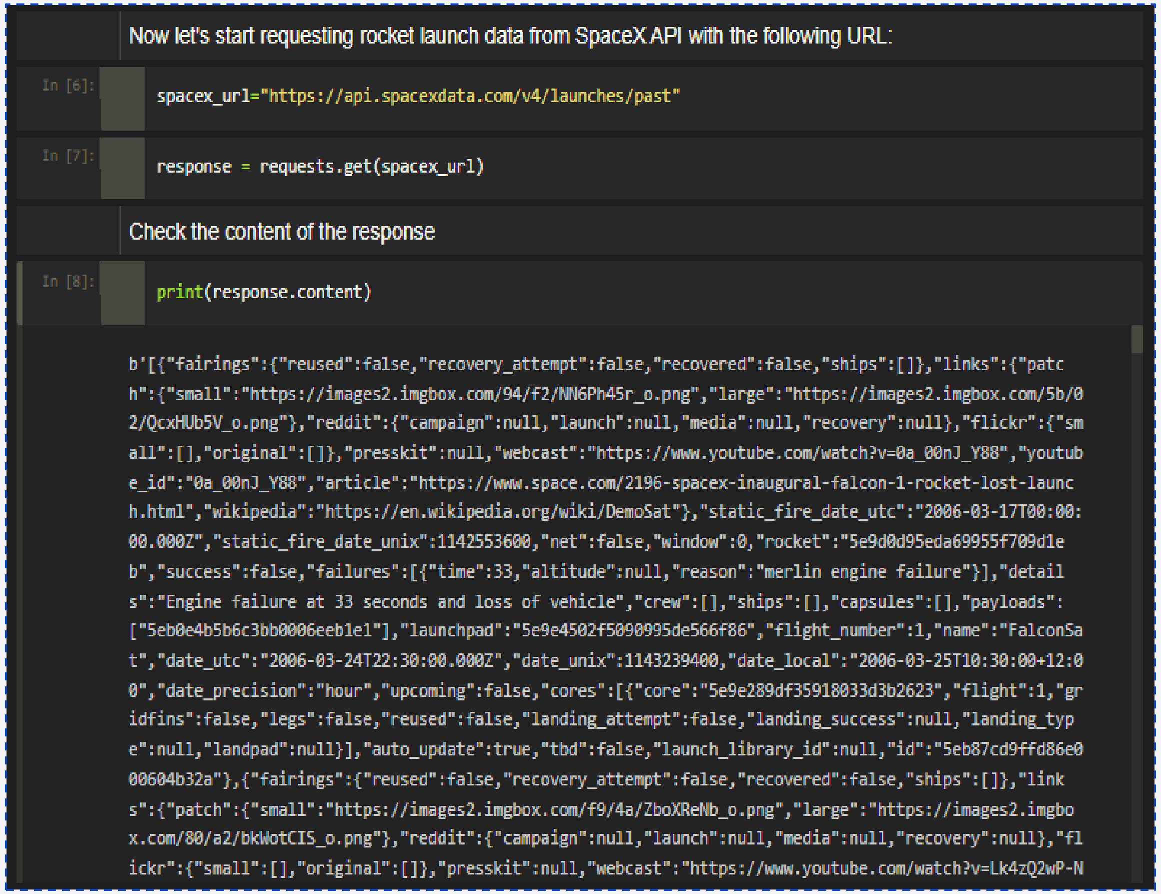
Data Collection

- SpaceX REST API was used in order to collect the required data, but it was partitioned in different URL’s and when directly downloaded it was encrypted.
- Nevertheless, the Dataset structure was identical to the data spotted in the SpaceX ‘Falcon 9’ Wikipedia Web Page.
- Hence, using the first resulting dataset as a skeleton, the web-scrapped data from Wikipedia was spotted accordingly.

Data Collection – SpaceX API

- As it can be seen, a lot of the content of the API is a string value with non easily readable data, which means it is clearly encrypted.

GitHub URL: [SpaceX API Calls Notebook]()



Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
In [6]:  spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]:  response = requests.get(spacex_url)
```

Check the content of the response

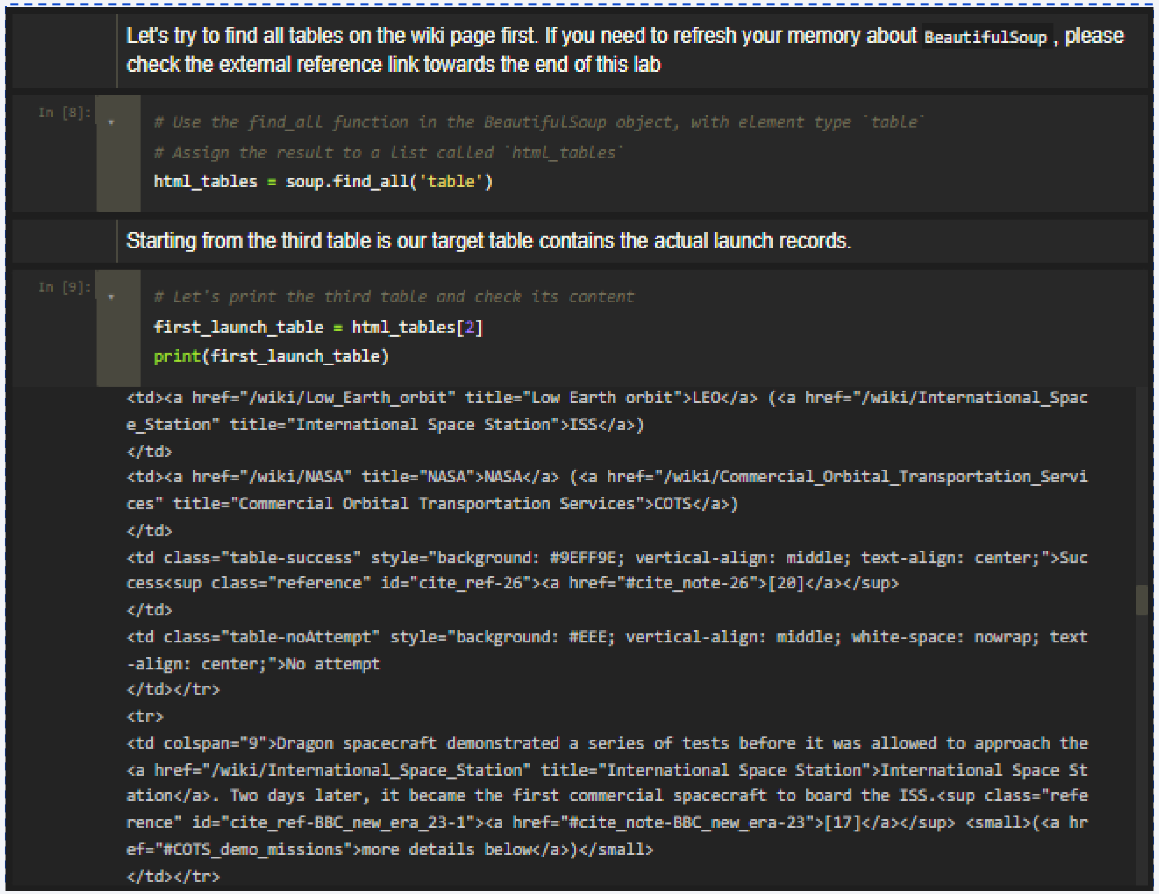
```
In [8]:  print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patc
h":{"small":"https://images2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/0
2/QcxHUb5V_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"sm
all":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00nJ_Y88","youtub
e_id":"0a_00nJ_Y88","article":"https://www.space.com/2196-spacex-inaugural-falcon-1-rocket-lost-launc
h.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:
00.000Z","static_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1e
b","success":false,"failures":[{"time":33,"altitude":null,"reason":"merlin engine failure"}],"detail
s":"Engine failure at 33 seconds and loss of vehicle","crew":[],"ships":[],"capsules":[],"payloads":
["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":1,"name":"FalconSa
t","date_utc":"2006-03-24T22:30:00.000Z","date_unix":1143239400,"date_local":"2006-03-25T10:30:00+12:0
0","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289df35918033d3b2623","flight":1,"gr
idfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_typ
e":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cd9ffd86e0
00604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"link
s":{"patch":{"small":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https://images2.imgbo
x.com/80/a2/bkWotCIS_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"fl
ickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=Lk4zQ2wP-N
```

Data Collection - Scraping

- Even if it is not completely readable, it can be spotted some kind of data structure in the response code.

GitHub URL: Web Scraping Notebook



Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup` , please check the external reference link towards the end of this lab

In [8]:

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

In [9]:

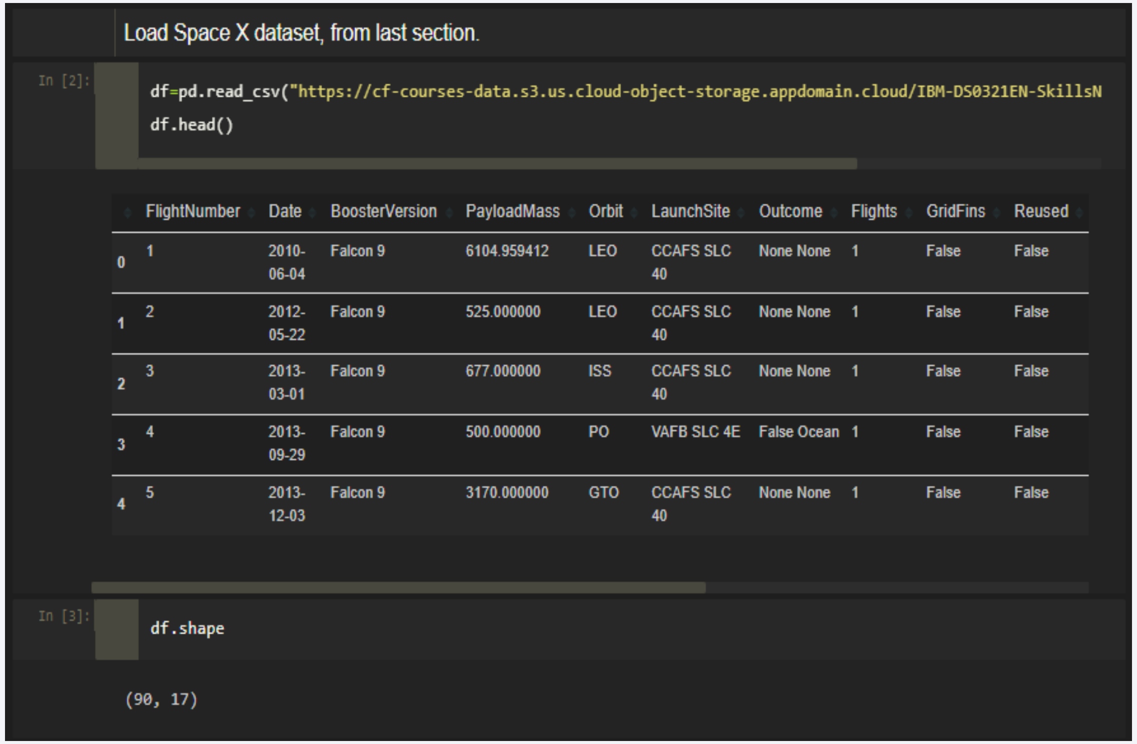
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<td><a href="/wiki/Low_Earth_orbit" title="Low Earth orbit">LEO</a> (<a href="/wiki/International_Spac
e_Station" title="International Space Station">ISS</a>)
</td>
<td><a href="/wiki/NASA" title="NASA">NASA</a> (<a href="/wiki/Commercial_Orbital_Transportation_Servi
ces" title="Commercial Orbital Transportation Services">COTS</a>)
</td>
<td class="table-success" style="background: #9EFF9E; vertical-align: middle; text-align: center;">Suc
cess<sup class="reference" id="cite_ref-26"><a href="#cite_note-26">[20]</a></sup>
</td>
<td class="table-noAttempt" style="background: #EEE; vertical-align: middle; white-space: nowrap; text
-align: center;">No attempt
</td></tr>
<tr>
<td colspan="9">Dragon spacecraft demonstrated a series of tests before it was allowed to approach the
<a href="/wiki/International_Space_Station" title="International Space Station">International Space St
ation</a>. Two days later, it became the first commercial spacecraft to board the ISS.<sup class="refe
rence" id="cite_ref-BBC_new_era_23-1"><a href="#cite_note-BBC_new_era-23">[17]</a></sup> <small>(<a hr
ef="#COTS_demo_missions">more details below</a>)</small>
</td></tr>
```

Data Wrangling

- Once the data is gathered and organized, we got a Pandas Dataframe.
- After that, data was sampled to only work with ‘Falcon 9’ Booster Version.
- Null Values were solved by replacing numerical data with the mean.
- Categorical values like Orbit, Outcome and Class were standardized.

GitHub URL: Data Wrangling Notebook



Load Space X dataset, from last section.

```
In [2]:
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsN
df.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False

```
In [3]:
df.shape
```

```
(90, 17)
```

EDA with Data Visualization

- In order to have a first approximation to the data, some visualizations were plotted for determining a notion over the success rate by making comparisons with the number of launches.
- This notion of the success rate was spotted with different plots that showed different relationships between variables like:
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Success Rate vs. Orbit Type
  - Flight Number vs. Orbit Type
  - Payload vs. Orbit Type
  - Launch Success Yearly Trend

GitHub URL: [EDA with Data Visualization Notebook]

EDA with SQL

- In order to analyze in table format and get some data insights of the dataset, it was used PostgreSQL's queries for:
    - Determining exactly which were the Launch Sites.
    - Stablish the total mass carried by boosters launched by NASA.
    - Know the average mass carried by booster version F9 v1.1.
    - Point the date of the first successful landing in Drone Ship.
    - The name of boosters which had a success in ground pad and of a determined Payload mass.

EDA with SQL

- In order to analyze in table format and get some data insights of the dataset, it was used PostgreSQL's queries for:
  - The total success and failure of the mission.
  - The records of each successful landing outcome from 2017.
  - The rank count of landing outcomes between two given dates.

GitHub URL: EDA with SQL Notebook

Build an Interactive Map with Folium

- As a geospatial reference an interactive map was developed with folium, where the following objects were added:
  - Highlighted Circles Areas: For knowing exactly where the launches were done.
  - Cluster Markers of Outcome Launches: Determine a notion over the success landing rate for each launch sites.
  - Distance between a launch site to strategic proximities: To indirectly determine how many risk was SpaceX determined to get in case that something went wrong with the mission success considering the proximity to places that were habited or had important infrastructure.

GitHub URL: Interactive Map with Folium Notebook

Build a Dashboard with Plotly Dash

- In order to get more insights on the features and set down notions over successful landings, the following graphs were deployed in the Dashboard:
  - Total Launches by Site Pie Chart: In order to understand the proportion of launches made in each place.
  - Site Dropdown for the Pie Chart: To filter by Launch Site and assess the Landing Success Rate of each one.
  - Payload Mass vs Booster Version Scatter Plot: For finding a notion over the Landing Success Rate and the type of Booster used on each launch and how much mass the payload was charging when deployed.
  - Range Slider over Payload Mass: For visualizing with ease which mass range has the more successful rate and trying at the same time to see which Booster Version has better results.

GitHub URL: Plotly Dashboard Notebook

Predictive Analysis (Classification)

- The process for developing the Classification Models was the following:
  - Data was Standardized and splatted into Train and Test Sets.
  - A Grid Search object was created with all the possible combination of parameters in order to get the best performing ones for each algorithm.
  - Models were the fitted with the best performing parameters and their accuracy measures were calculated by using the test set.
  - Finally, a confusion matrix was built for visualizing the success classifications and types of error clarifications.
  - Summarize how you built, evaluated, improved, and found the best performing classification model

GitHub URL: Predictive Analysis Notebook

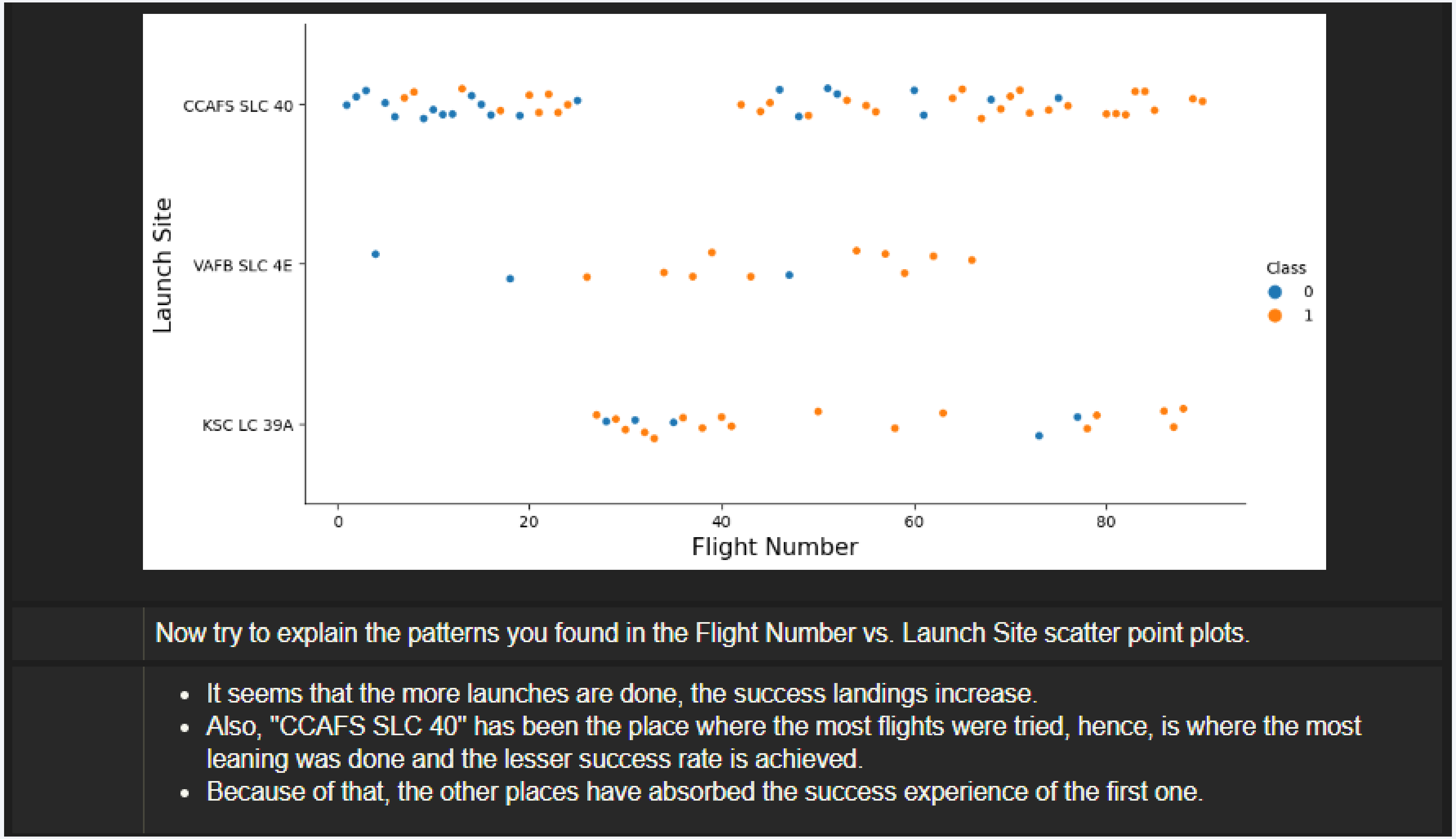
Results

- The learning curve toward successful landings seems to increase the success rate; That is, the more launches, the more controlled landings.
- There seem to be launch places determined to get a higher successful landing rate, clearly, missions whose outcomes are determined to be failures won’t be placed in a habited surrounding area.
- There seems to be a better performance with pure classifications algorithms like:
  - Decision Tree: Higher Precision
  - Support Vector Machine: Best Consistency

Section 2

Insights drawn from EDA

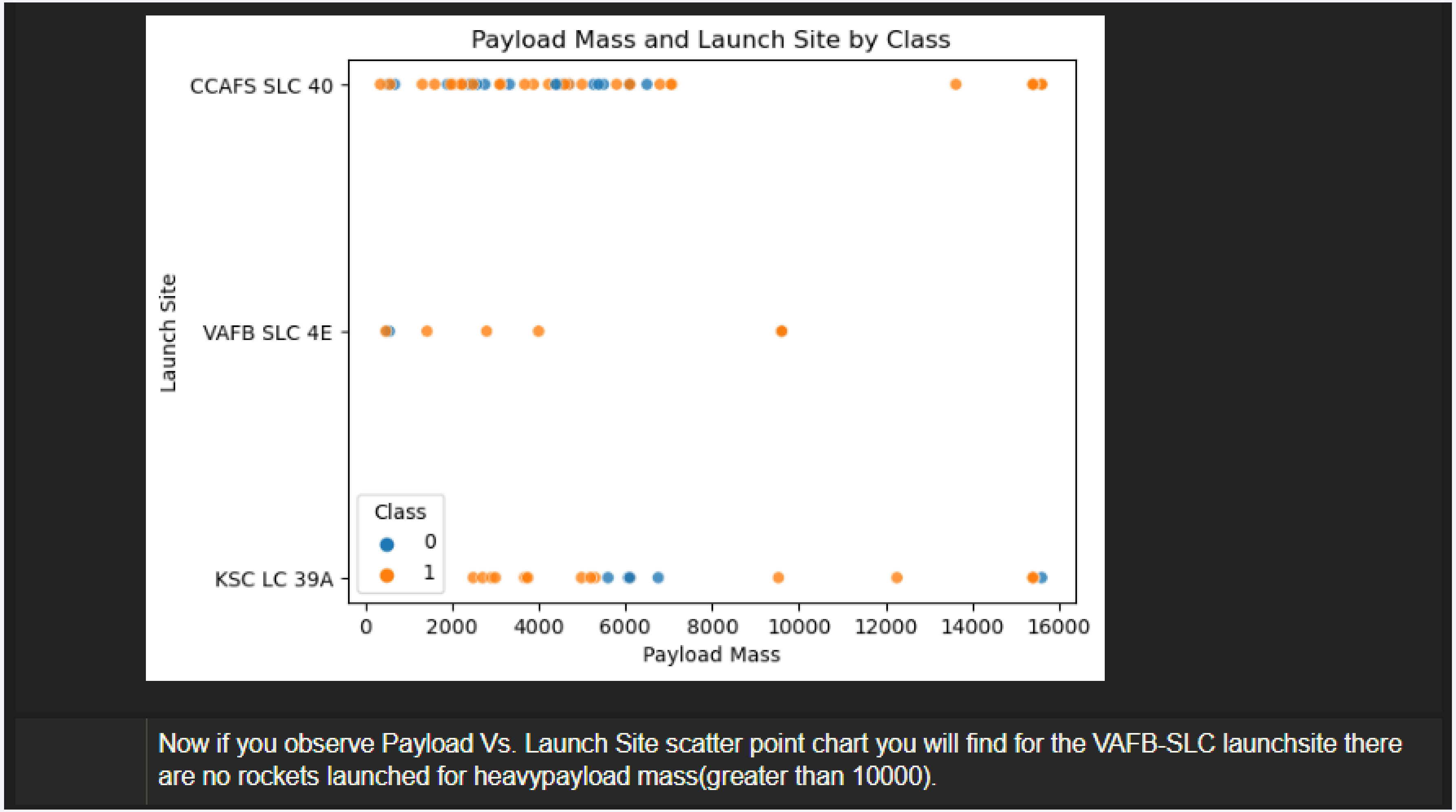
Flight Number vs. Launch Site



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

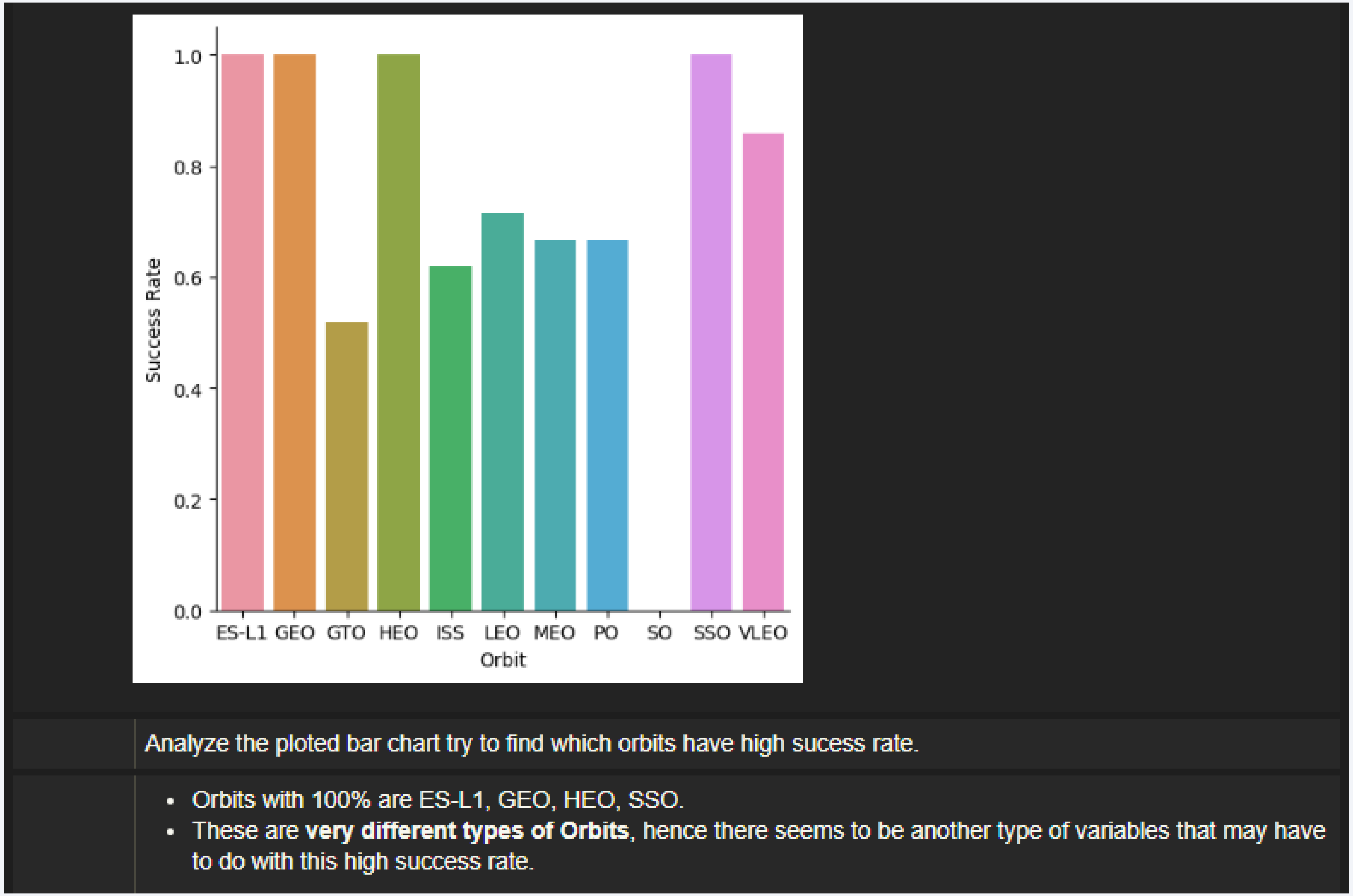
- It seems that the more launches are done, the success landings increase.
- Also, "CCAFS SLC 40" has been the place where the most flights were tried, hence, is where the most leaning was done and the lesser success rate is achieved.
- Because of that, the other places have absorbed the success experience of the first one.

Payload vs. Launch Site



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

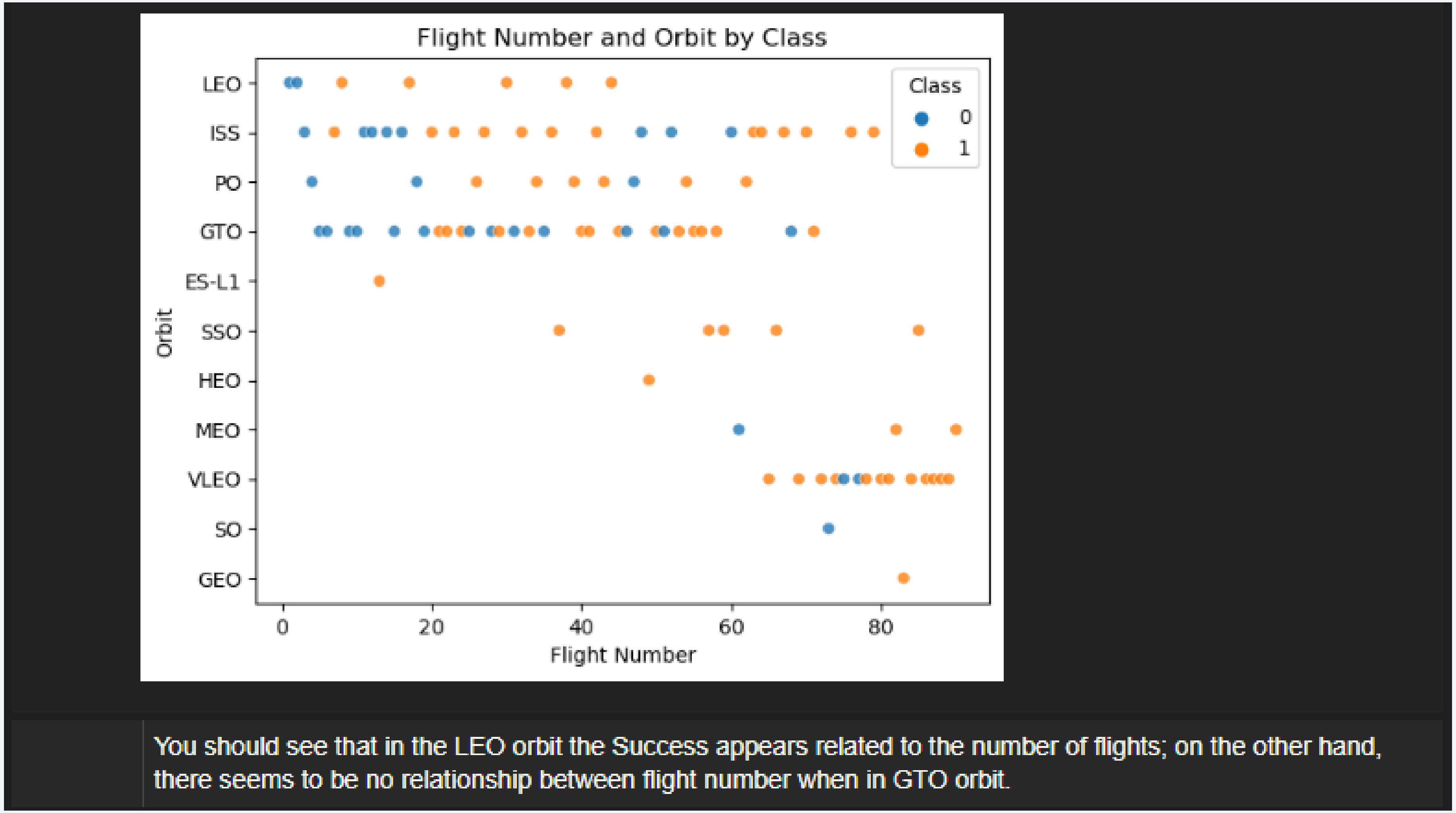
Success Rate vs. Orbit Type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

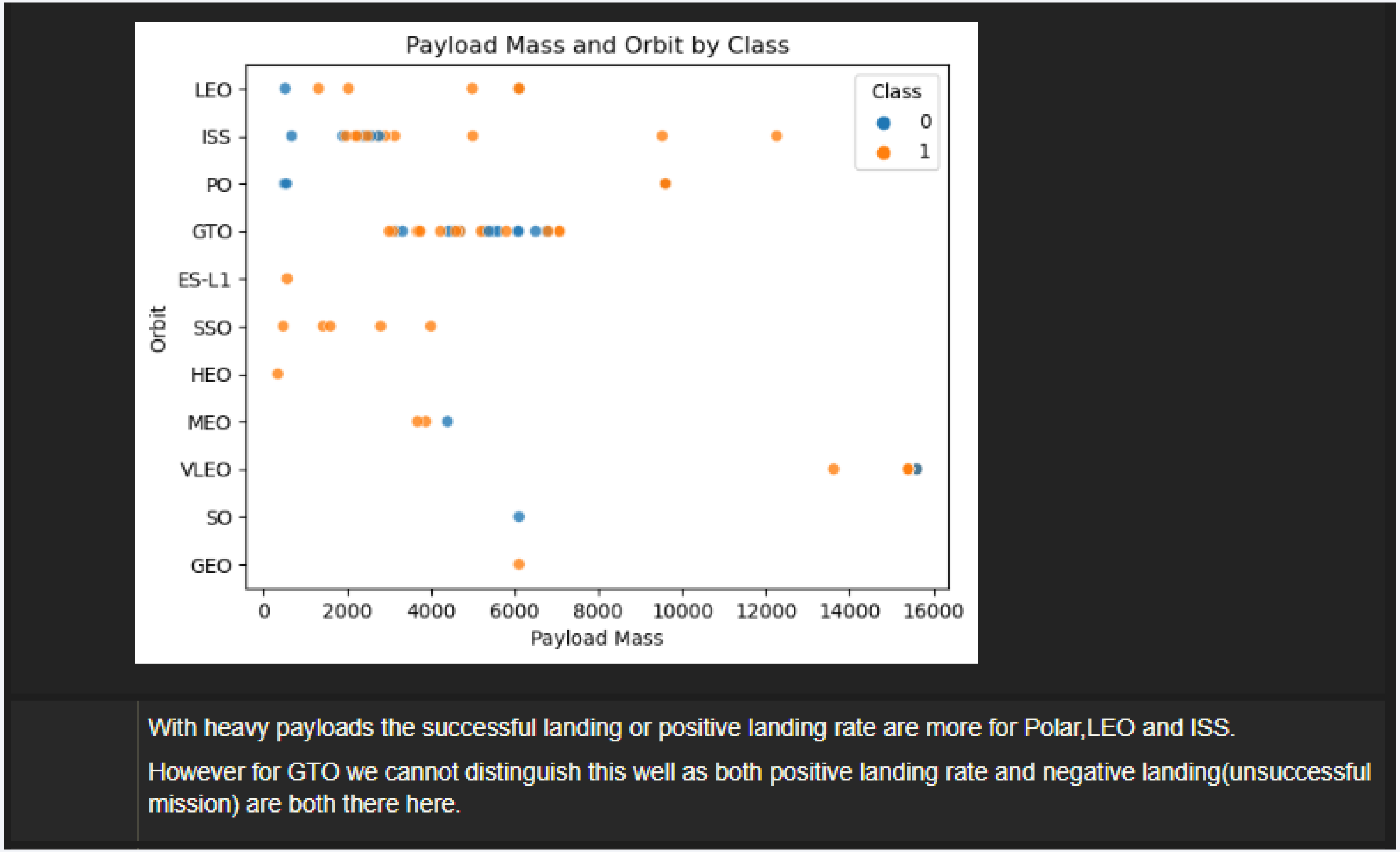
- Orbits with 100% are ES-L1, GEO, HEO, SSO.
- These are **very different types of Orbits**, hence there seems to be another type of variables that may have to do with this high success rate.

Flight Number vs. Orbit Type



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

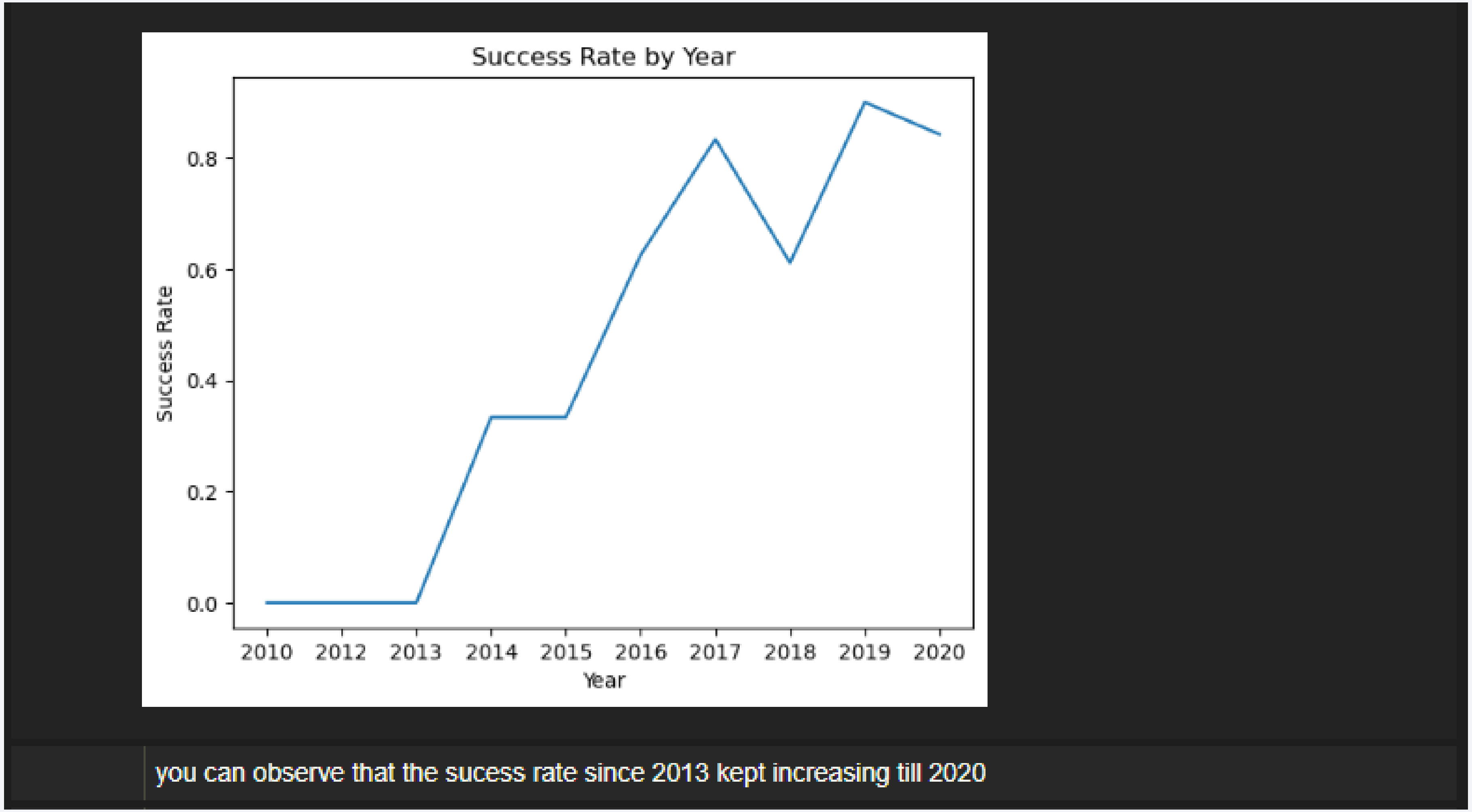
Payload Mass vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

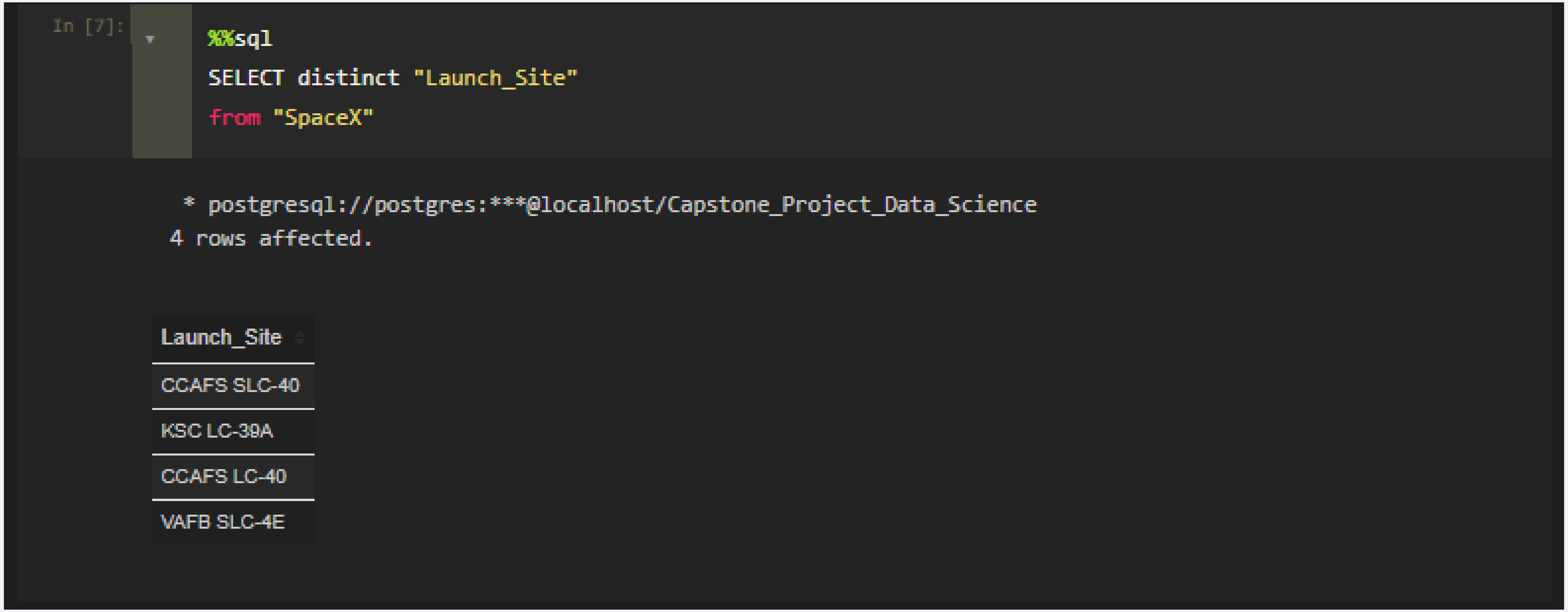
However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names



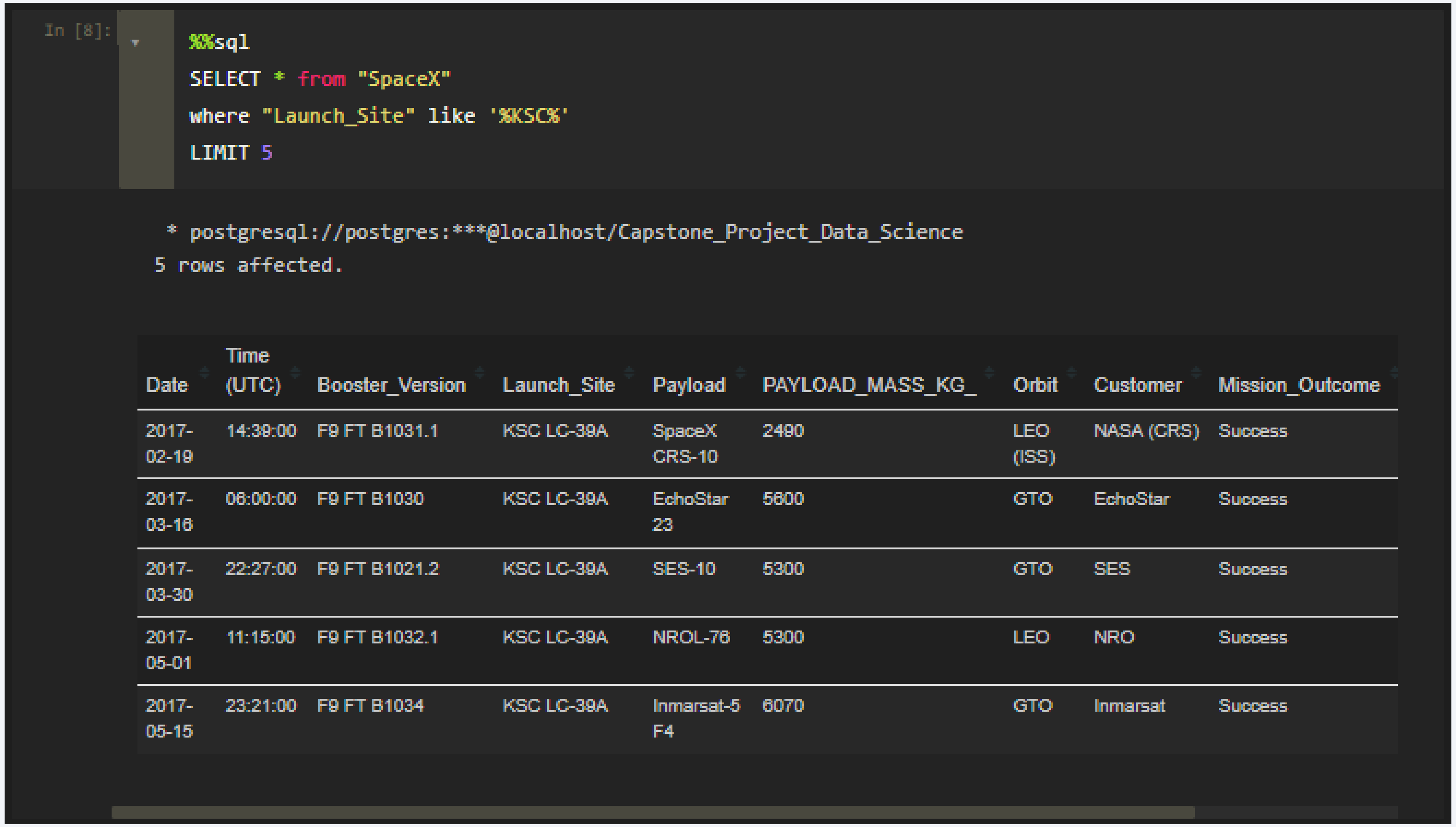
```
%%sql
SELECT distinct "Launch_Site"
from "SpaceX"
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
4 rows affected.
```

Launch_Site
CCAFS SLC-40
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E

- The query distinguished between all the launch site's values and made a list over it.

Launch Site Names Begin with 'KSC'



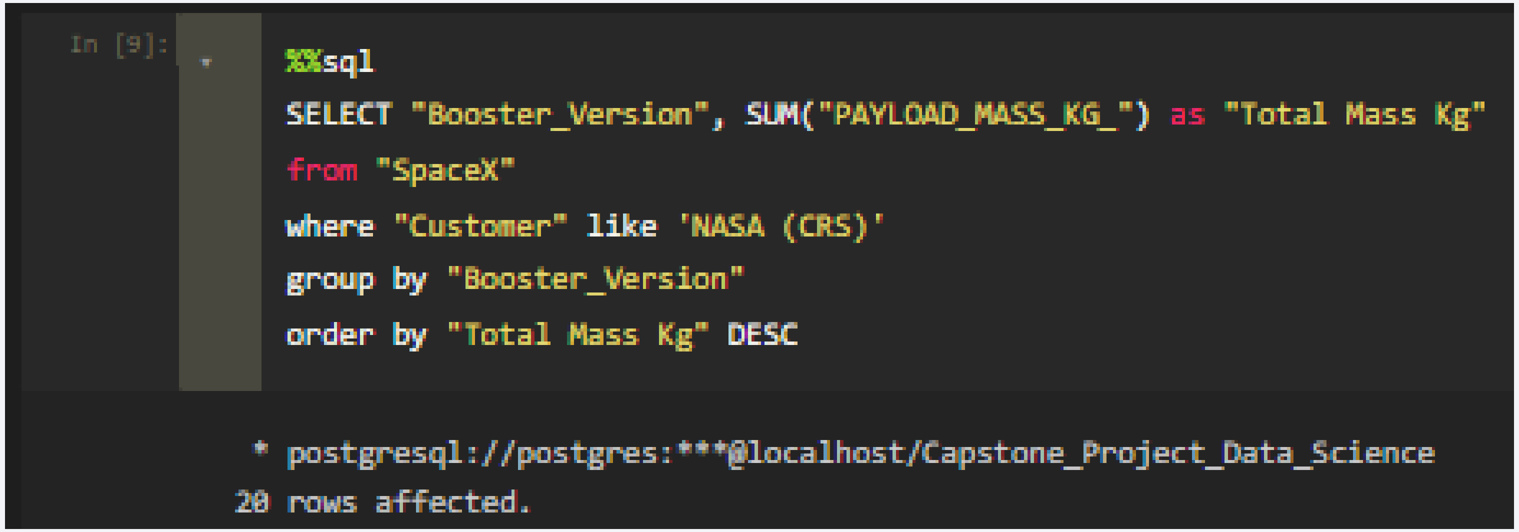
```
%%sql
SELECT * from "SpaceX"
where "Launch_Site" like '%KSC%'
LIMIT 5
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
5 rows affected.
```

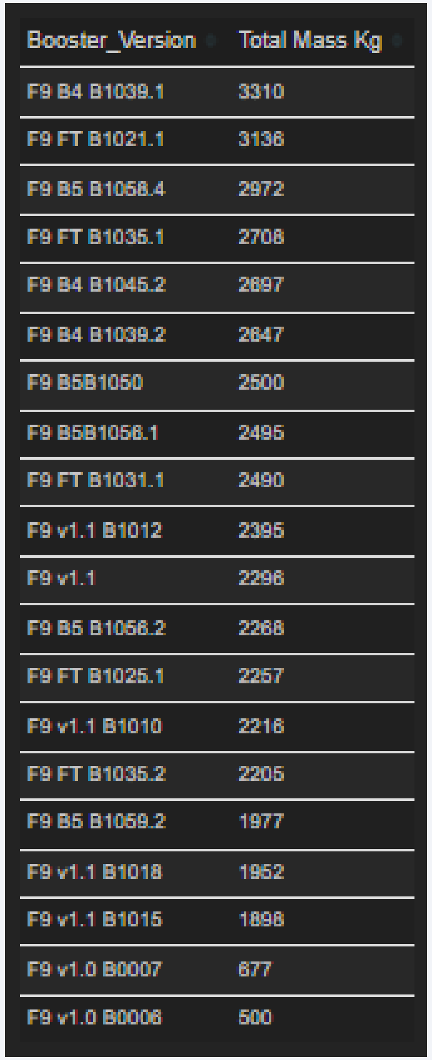
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success

- The query filtered over the launch site value ‘KSC’ and returned the first 5 observations. It can be seen that the first launches were in 2017.

Total Payload Mass



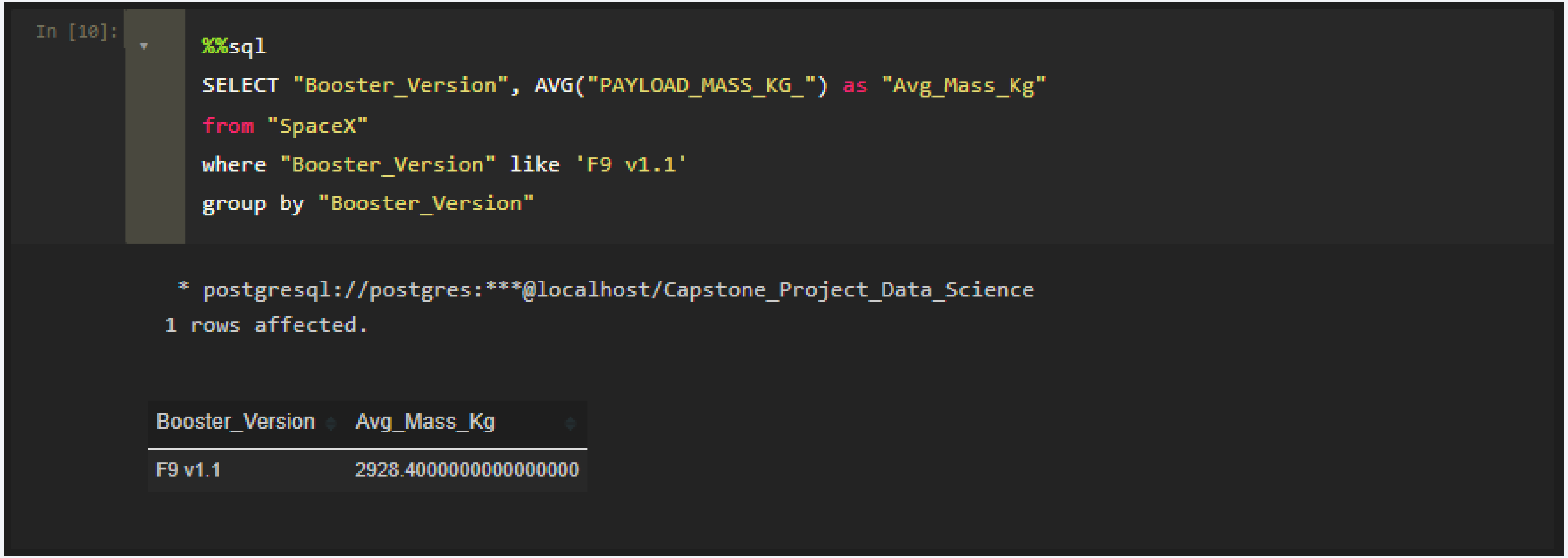
```
In [9]:
%%sql
SELECT "Booster_Version", SUM("PAYLOAD_MASS_KG_") as "Total Mass Kg"
from "SpaceX"
where "Customer" like 'NASA (CRS)'
group by "Booster_Version"
order by "Total Mass Kg" DESC

 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
20 rows affected.
```



Booster_Version	Total Mass Kg
F9 B4 B1039.1	3310
F9 FT B1021.1	3136
F9 B5 B1058.4	2972
F9 FT B1035.1	2708
F9 B4 B1045.2	2697
F9 B4 B1039.2	2647
F9 B5B1050	2500
F9 B5B1056.1	2495
F9 FT B1031.1	2490
F9 v1.1 B1012	2395
F9 v1.1	2296
F9 B5 B1056.2	2268
F9 FT B1025.1	2257
F9 v1.1 B1010	2216
F9 FT B1035.2	2205
F9 B5 B1059.2	1977
F9 v1.1 B1018	1952
F9 v1.1 B1015	1898
F9 v1.0 B0007	677
F9 v1.0 B0006	500

- The query makes a summatory of the mass carried by the payloads during the whole study period and done by NASA.

Average Payload Mass by F9 v1.1



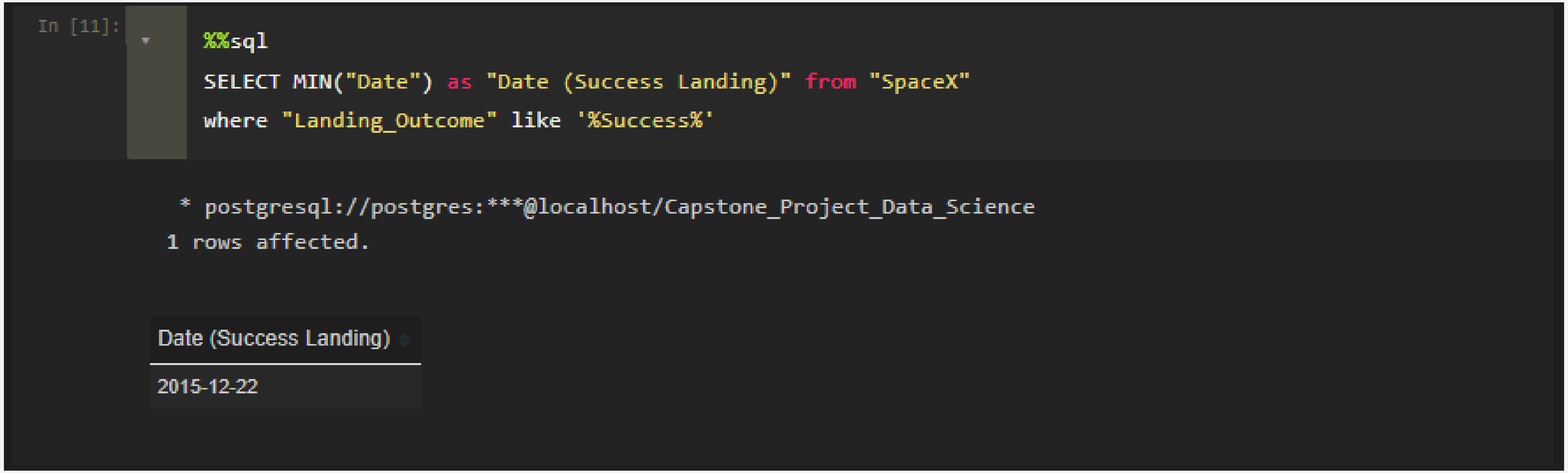
```
%%sql
SELECT "Booster_Version", AVG("PAYLOAD_MASS_KG_") as "Avg_Mass_Kg"
from "SpaceX"
where "Booster_Version" like 'F9 v1.1'
group by "Booster_Version"
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
1 rows affected.
```

Booster_Version	Avg_Mass_Kg
F9 v1.1	2928.4000000000000000

- The query shows the average payload mass carried by the F9 v1.1. Booster

First Successful Ground Landing Date



In [11]:

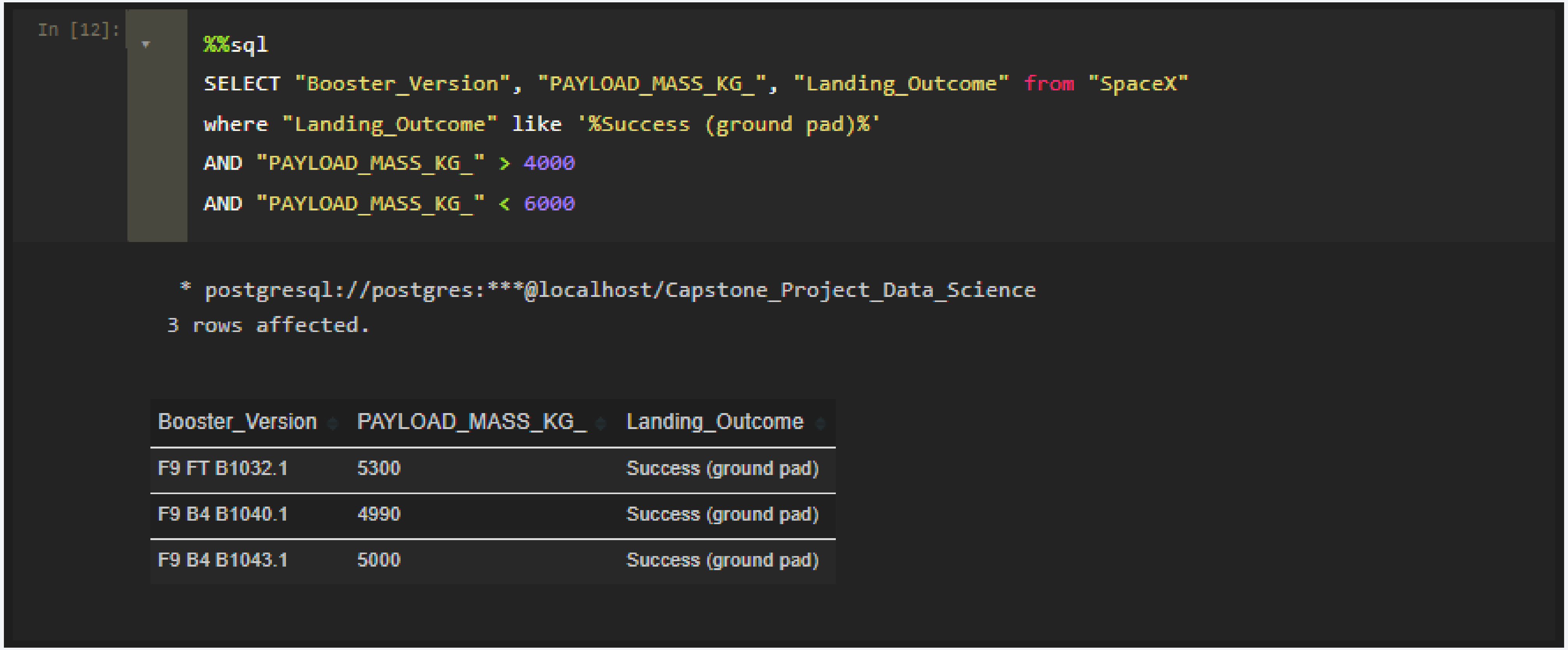
```
%%sql
SELECT MIN("Date") as "Date (Success Landing)" from "SpaceX"
where "Landing_Outcome" like '%Success%'
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
1 rows affected.
```

Date (Success Landing)
2015-12-22

- The query search for the minimum date that has a Successful Landing Outcome

Successful Drone Ship Landing with Payload between 4000 and 6000



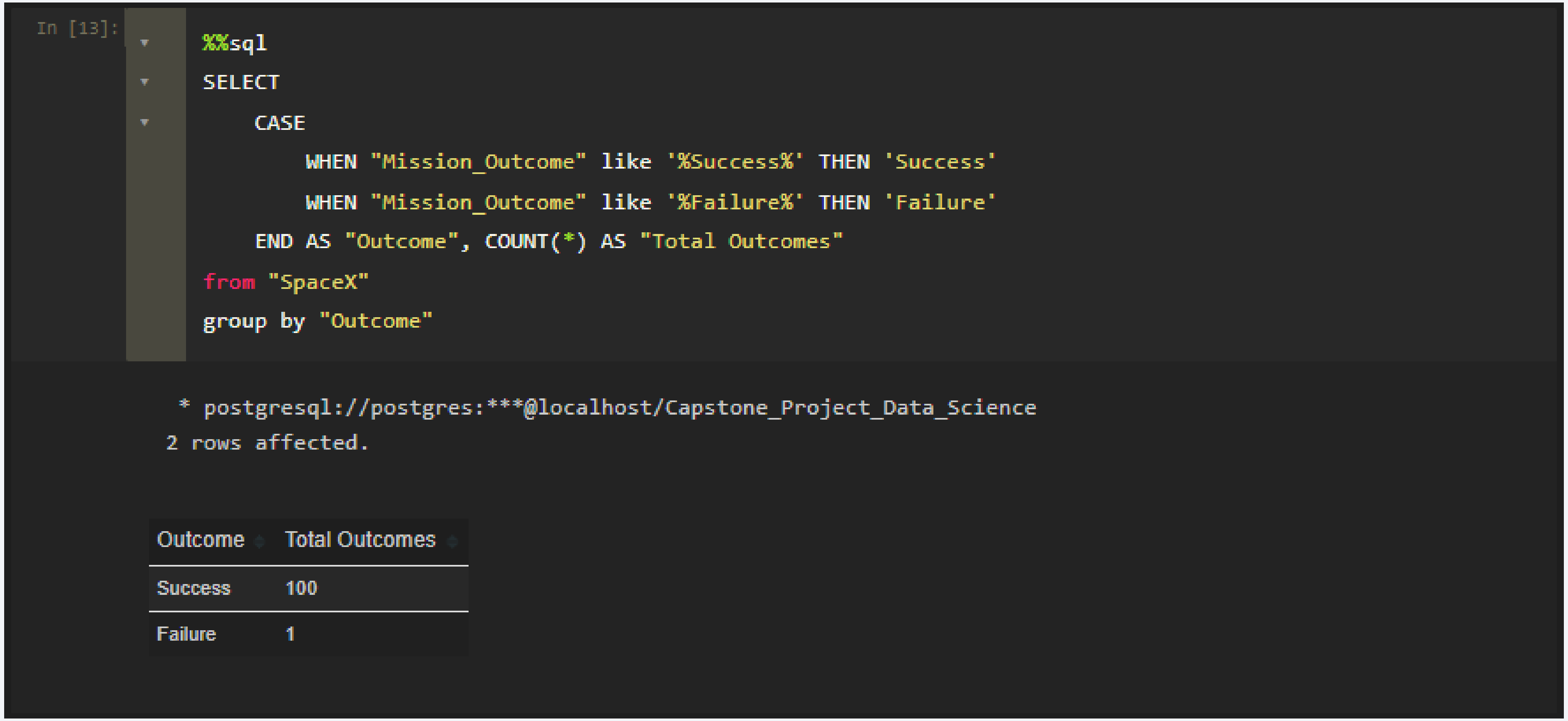
```
%%sql
SELECT "Booster_Version", "PAYLOAD_MASS_KG_", "Landing_Outcome" from "SpaceX"
where "Landing_Outcome" like '%Success (ground pad)%'
AND "PAYLOAD_MASS_KG_" > 4000
AND "PAYLOAD_MASS_KG_" < 6000
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
3 rows affected.
```

Booster_Version	PAYLOAD_MASS_KG_	Landing_Outcome
F9 FT B1032.1	5300	Success (ground pad)
F9 B4 B1040.1	4990	Success (ground pad)
F9 B4 B1043.1	5000	Success (ground pad)

- The query deploys data from the booster versions that were able to successfully land and carry the given mass range.

Total Number of Successful and Failure Mission Outcomes



In [13]:

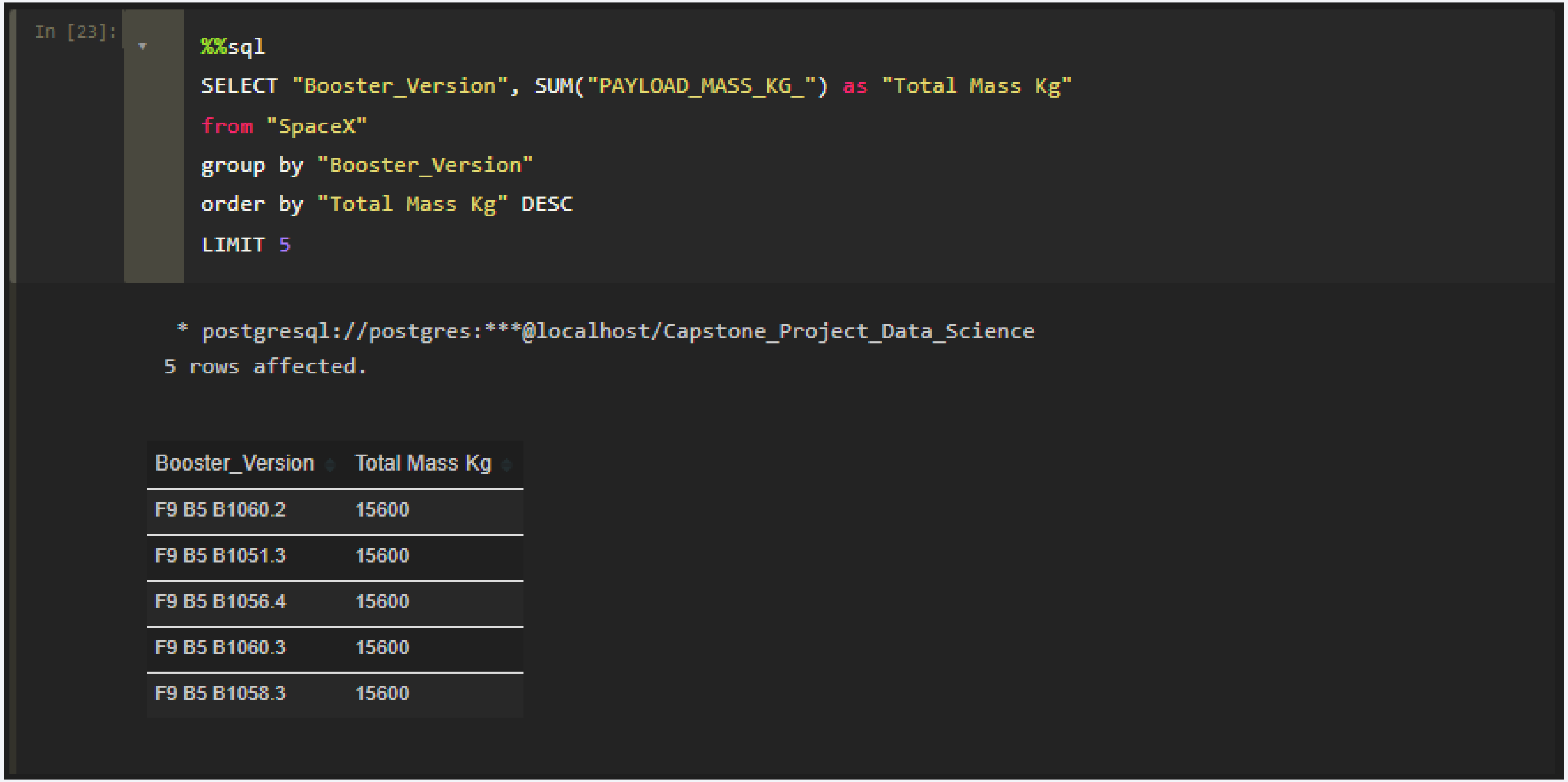
```
%%sql
SELECT
    CASE
        WHEN "Mission_Outcome" like '%Success%' THEN 'Success'
        WHEN "Mission_Outcome" like '%Failure%' THEN 'Failure'
    END AS "Outcome", COUNT(*) AS "Total Outcomes"
from "SpaceX"
group by "Outcome"
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
2 rows affected.
```

Outcome	Total Outcomes
Success	100
Failure	1

- The query presents the mission outcome standardized and grouped
- These results implies is important to determine common features among missions which landing outcomes are failures

Boosters Carried Maximum Payload



In [23]:

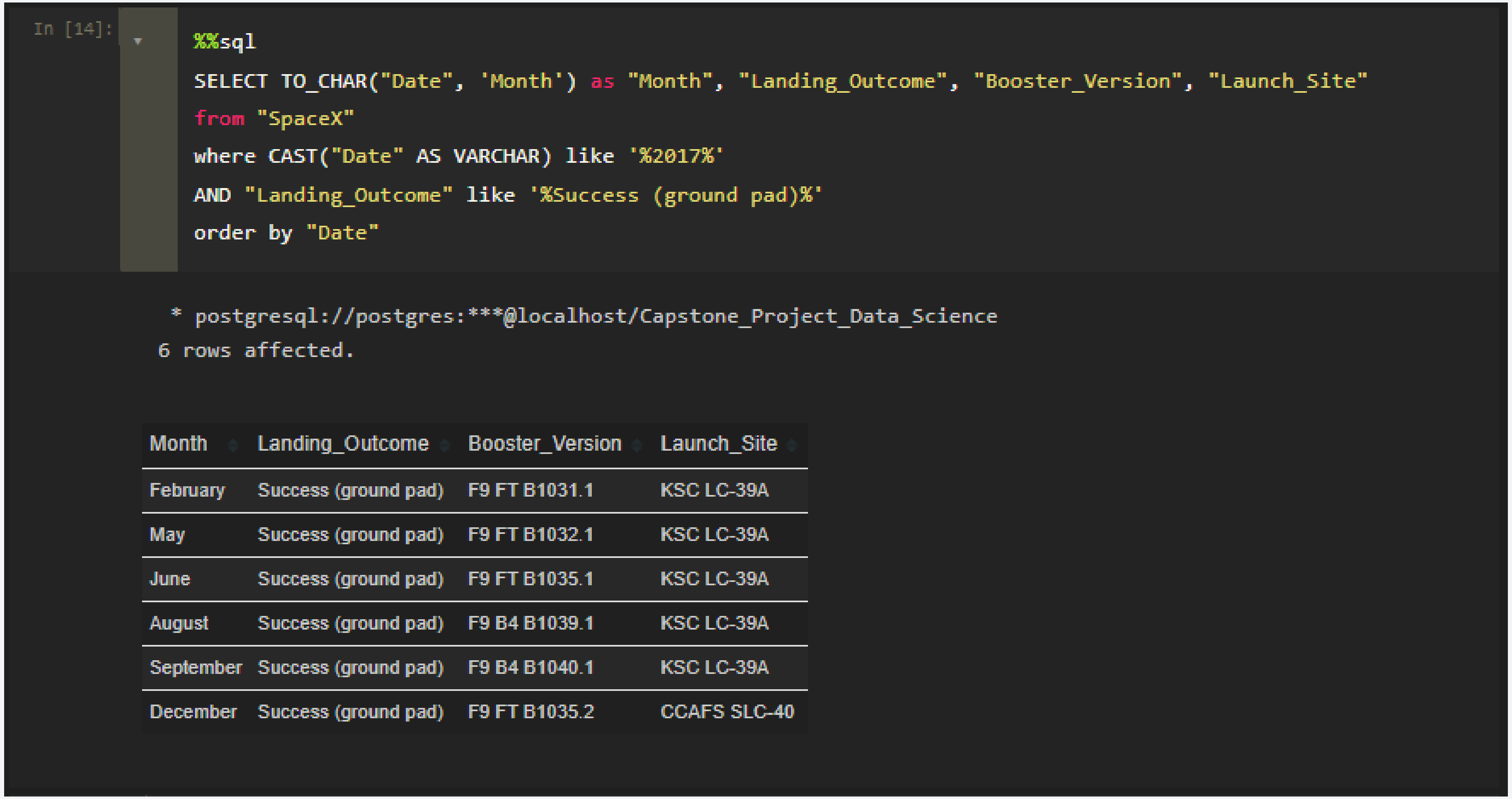
```
%%sql
SELECT "Booster_Version", SUM("PAYLOAD_MASS_KG_") as "Total Mass Kg"
from "SpaceX"
group by "Booster_Version"
order by "Total Mass Kg" DESC
LIMIT 5
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
5 rows affected.
```

Booster_Version	Total Mass Kg
F9 B5 B1060.2	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1060.3	15600
F9 B5 B1058.3	15600

- The query shows the total mass accumulated carried by each Booster Version

2017 Launch Records



In [14]:

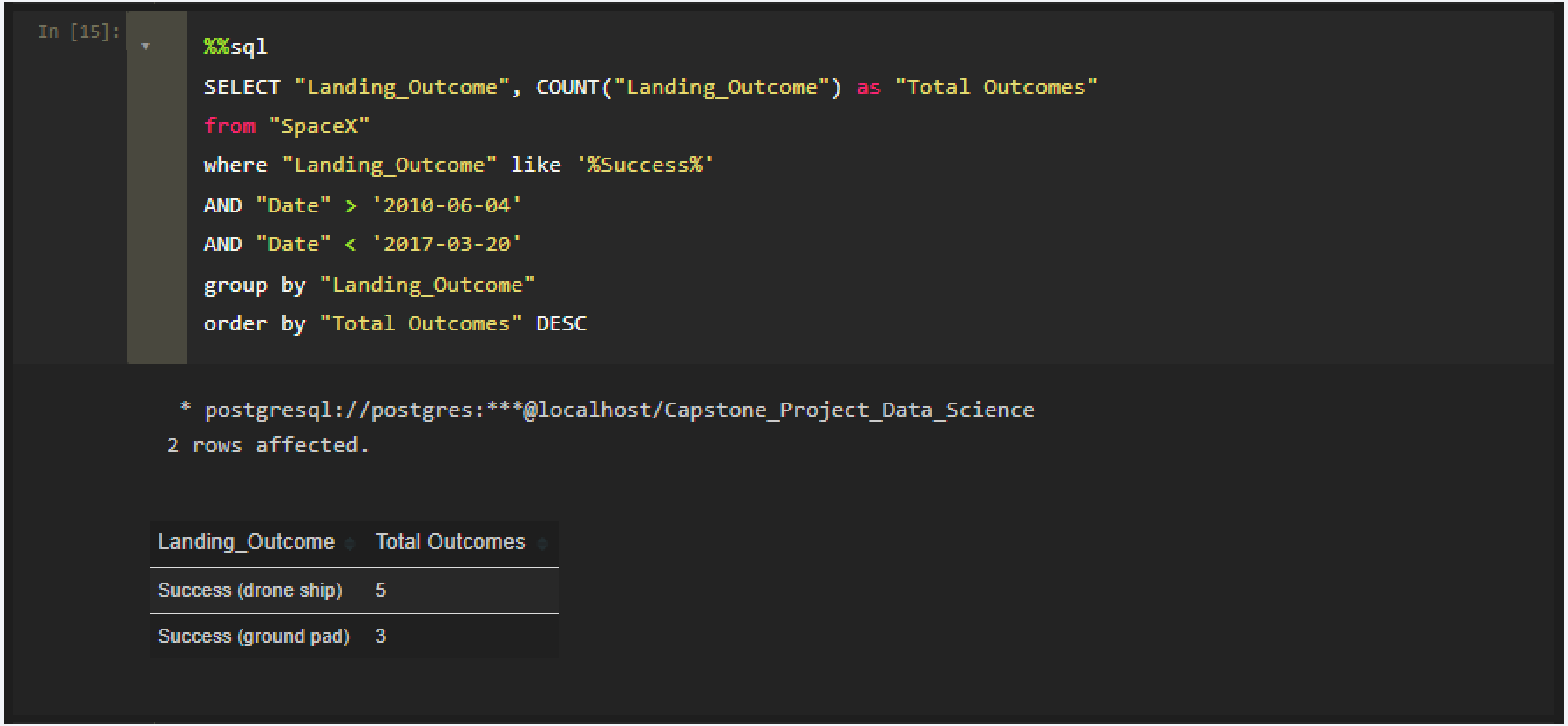
```
%%sql
SELECT TO_CHAR("Date", 'Month') as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site"
from "SpaceX"
where CAST("Date" AS VARCHAR) like '%2017%'
AND "Landing_Outcome" like '%Success (ground pad)%'
order by "Date"
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
6 rows affected.
```

Month	Landing_Outcome	Booster_Version	Launch_Site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

- The query prepares the data to be separated by months which is done by specifying the year and the landing outcome of interest

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20



In [15]:

```
%%sql
SELECT "Landing_Outcome", COUNT("Landing_Outcome") as "Total Outcomes"
from "SpaceX"
where "Landing_Outcome" like '%Success%'
AND "Date" > '2010-06-04'
AND "Date" < '2017-03-20'
group by "Landing_Outcome"
order by "Total Outcomes" DESC
```

```
 * postgresql://postgres:***@localhost/Capstone_Project_Data_Science
2 rows affected.
```

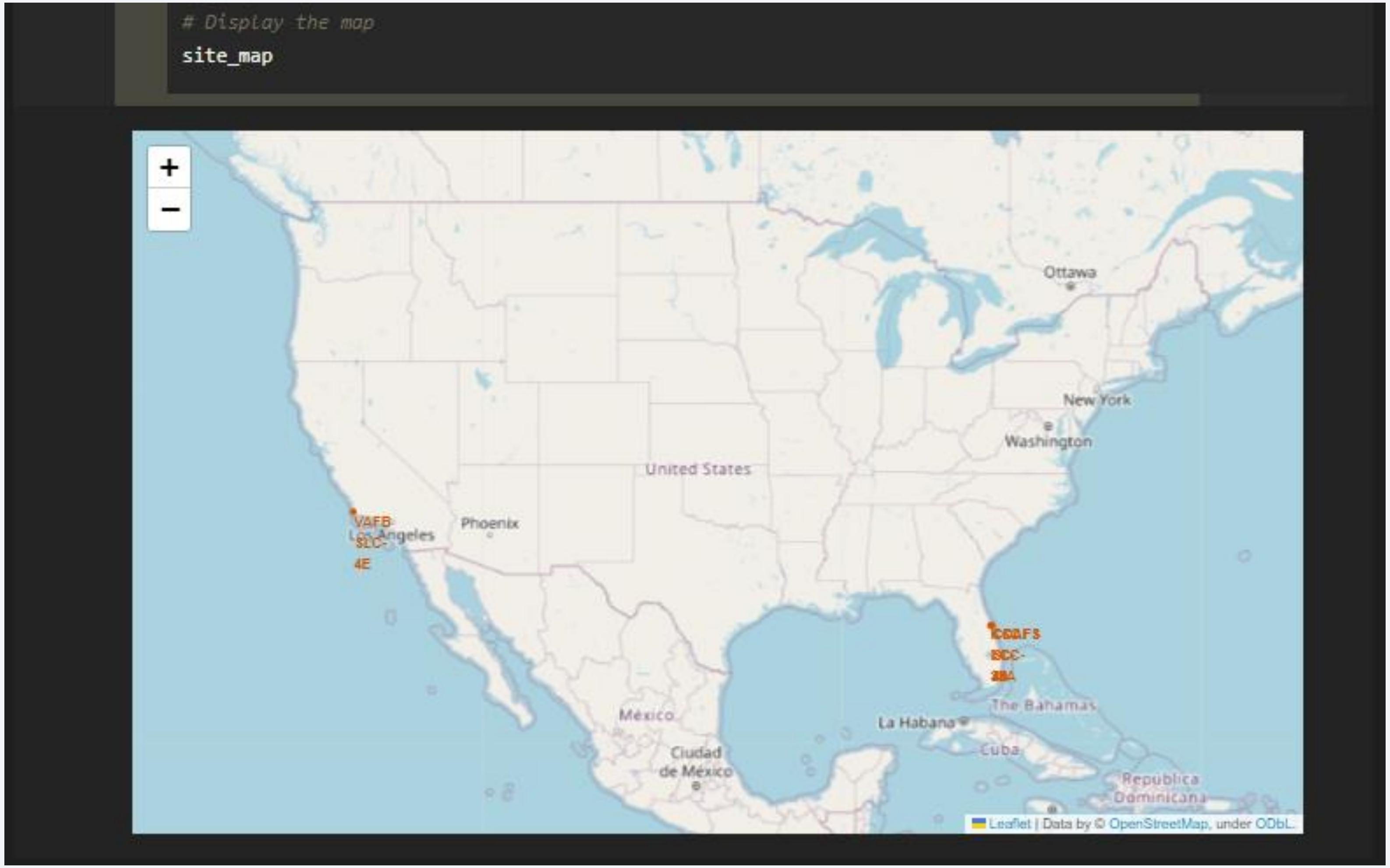
Landing_Outcome	Total Outcomes
Success (drone ship)	5
Success (ground pad)	3

- The query counts the successful landing outcomes and delimiting the date range by the one of interest.

Section 3

Launch Sites Proximities Analysis

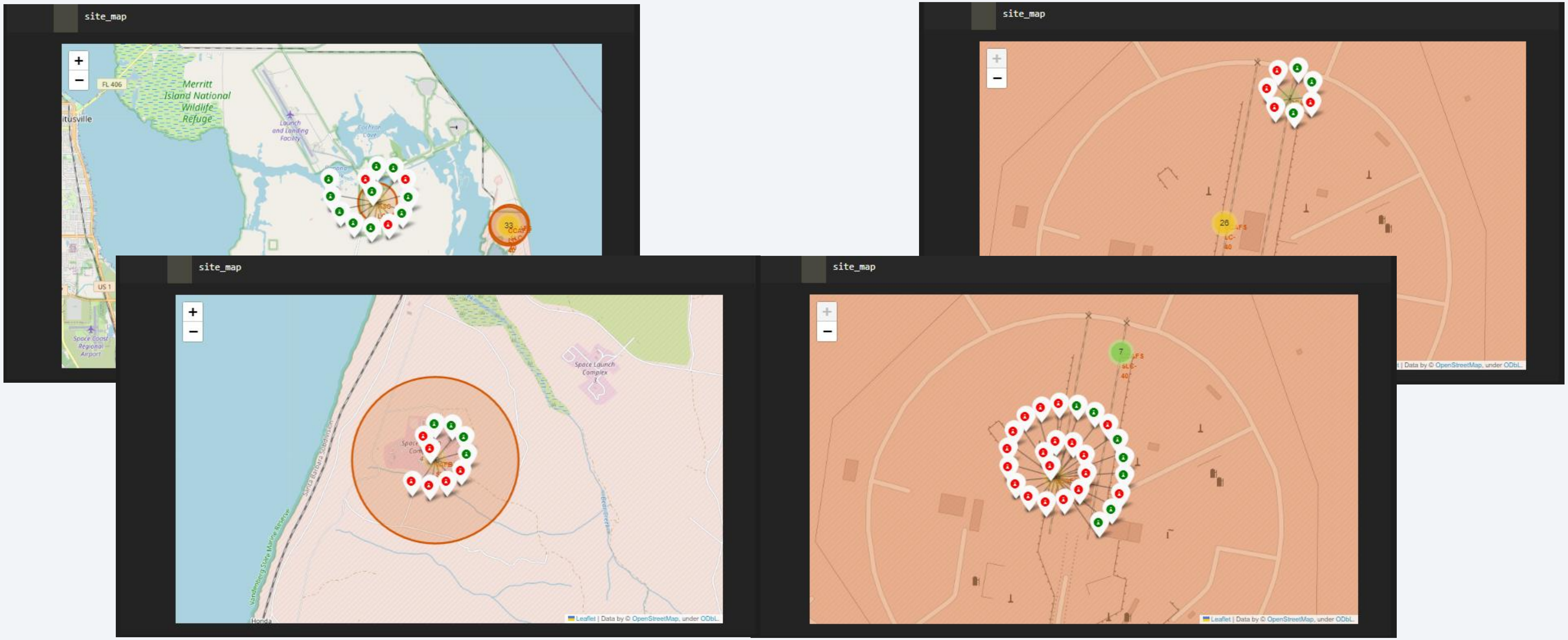
Launch Sites



```
# Display the map
site_map
```

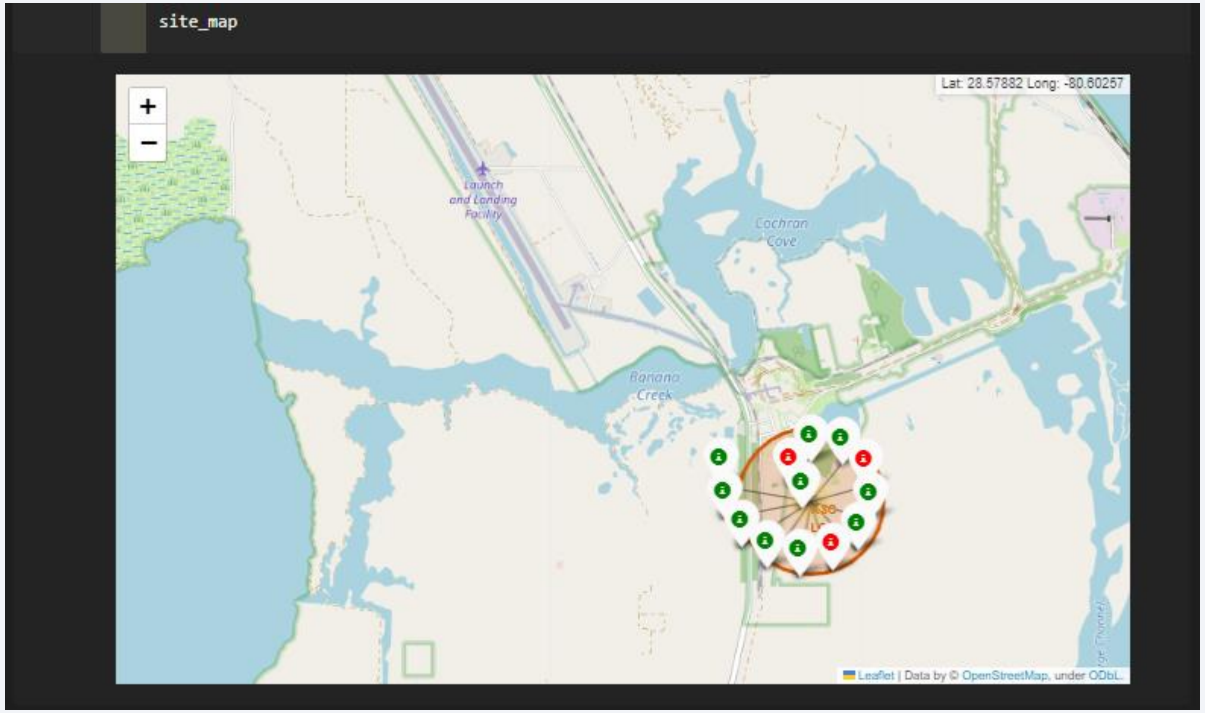
- The map suggest that such dramatic difference with one of the launch sites that serve to different purposes

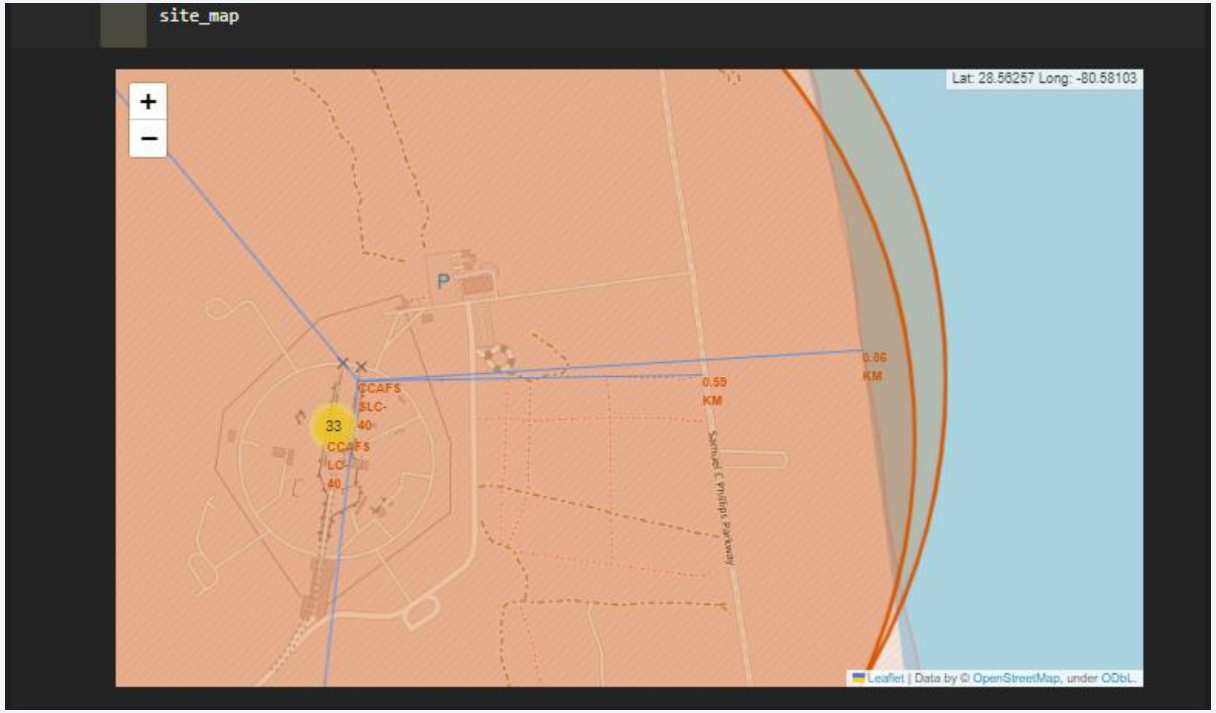
Notions of Successful Landing Rate by Launch Sites


site_map
site_map
site_map
site_map
+
−
FL 406
Merritt Island National Wildlife Refuge
itusville
Launch and Landing Facility
33
US 1
Space Coast Regional Airport
28
Space Launch Complex
7
Leaflet | Data by © OpenStreetMap, under ODbL.

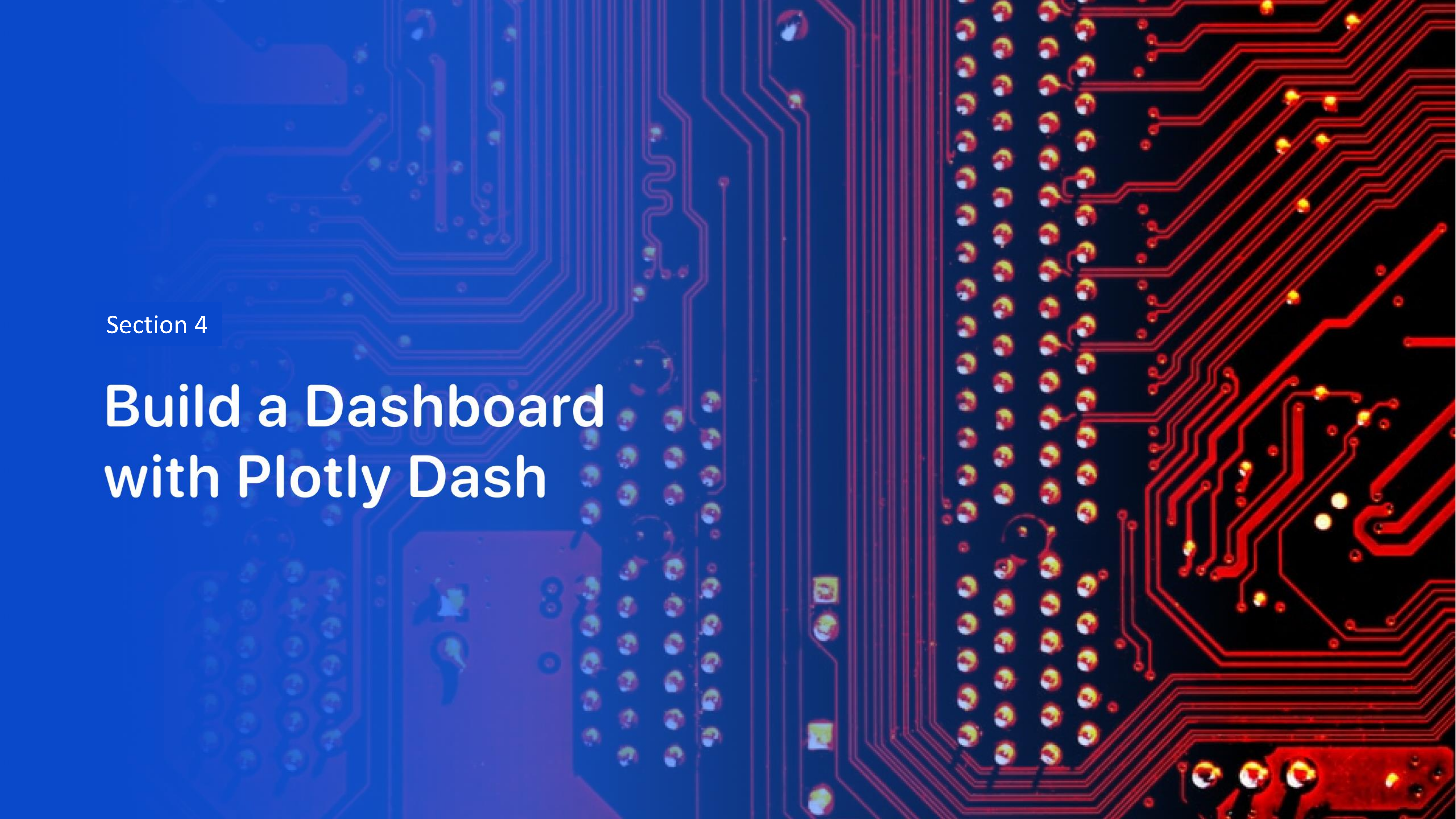
- Except for the upper left launch site, the rest of places are used frequently for failing the landings.
- That specific launch site is the one more surrounded by civil infraestructure.

Proximities to Launch Site


site_map
+
−
Lat: 28.57882 Long: -80.60257
Launch and Landing Facility
Cochran Cove
Banana Creek
Leaflet | Data by © OpenStreetMap, under ODbL.


site_map
+
−
Lat: 28.56257 Long: -80.58103
P
CCAFS SLC-40
33
CCAFS LC-40
0.59 KM
0.86 KM
Samuel C Phillips Parkway
Leaflet | Data by © OpenStreetMap, under ODbL.

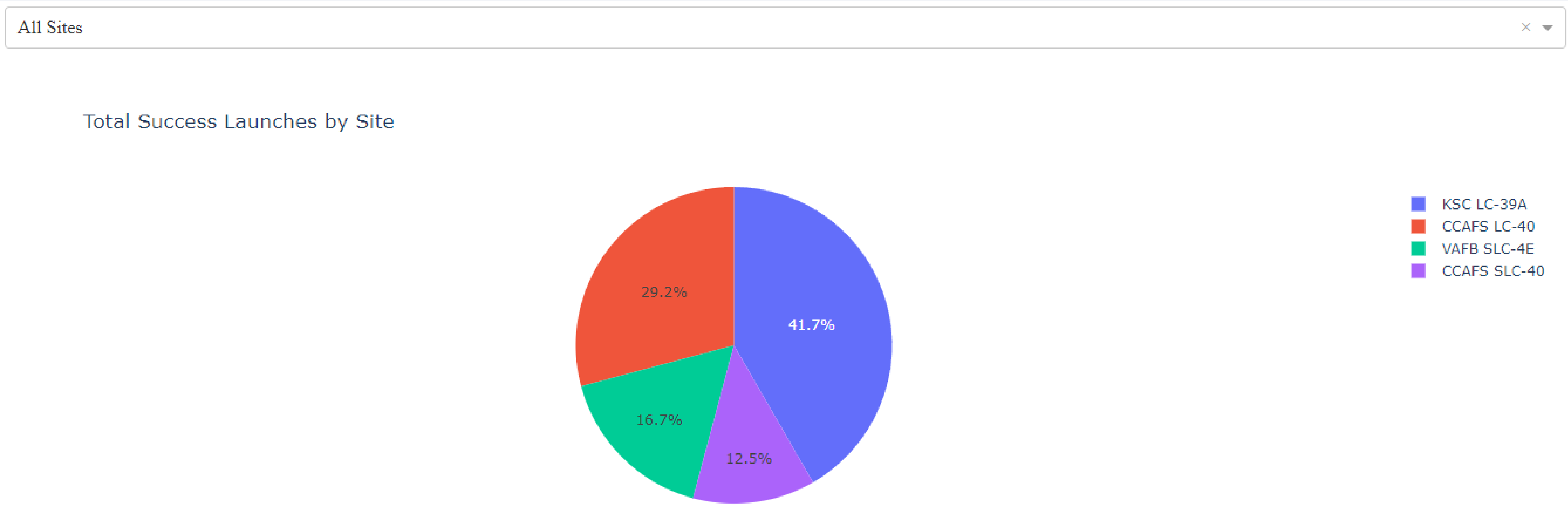
- As seen, the orange circle is a useful parameter to visually estimate distances.
- Nevertheless, it’s clear to see that left side launch site is the most surrounded by important infrastructure.



Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard: Launches by Sites

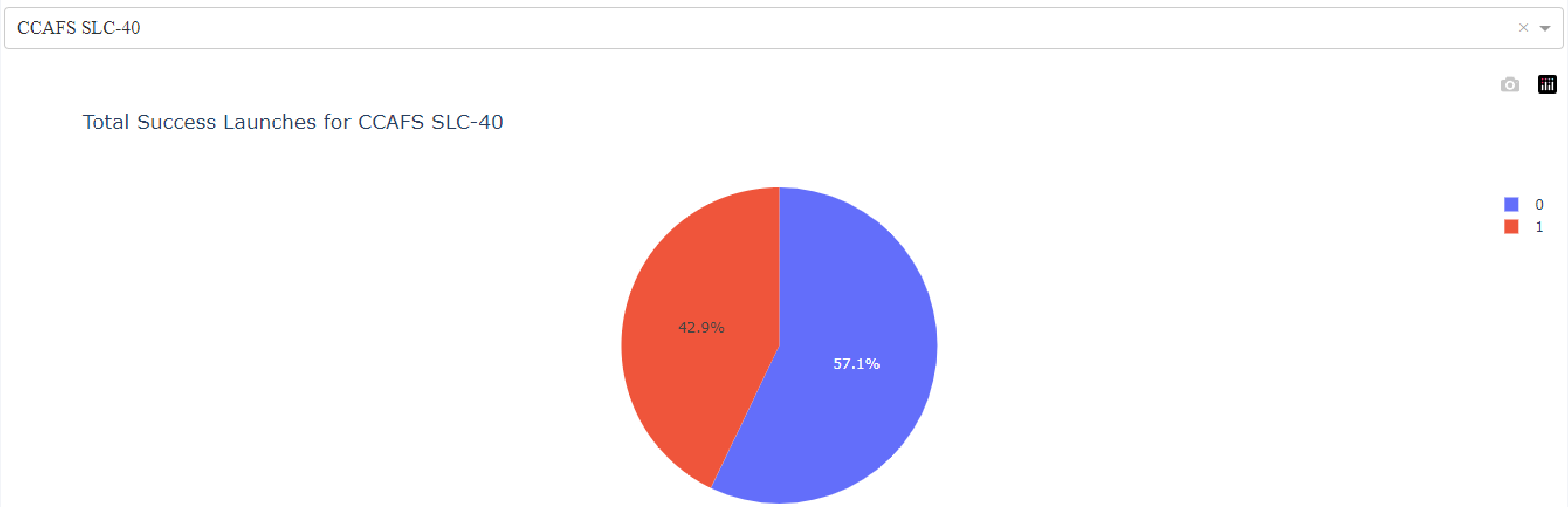


All Sites

- The place with the biggest proportion of launches are:

1. KSC LC-39A
2. CCAFS LC-40

SpaceX Launch Records Dashboard: Biggest Successful Launches Rate

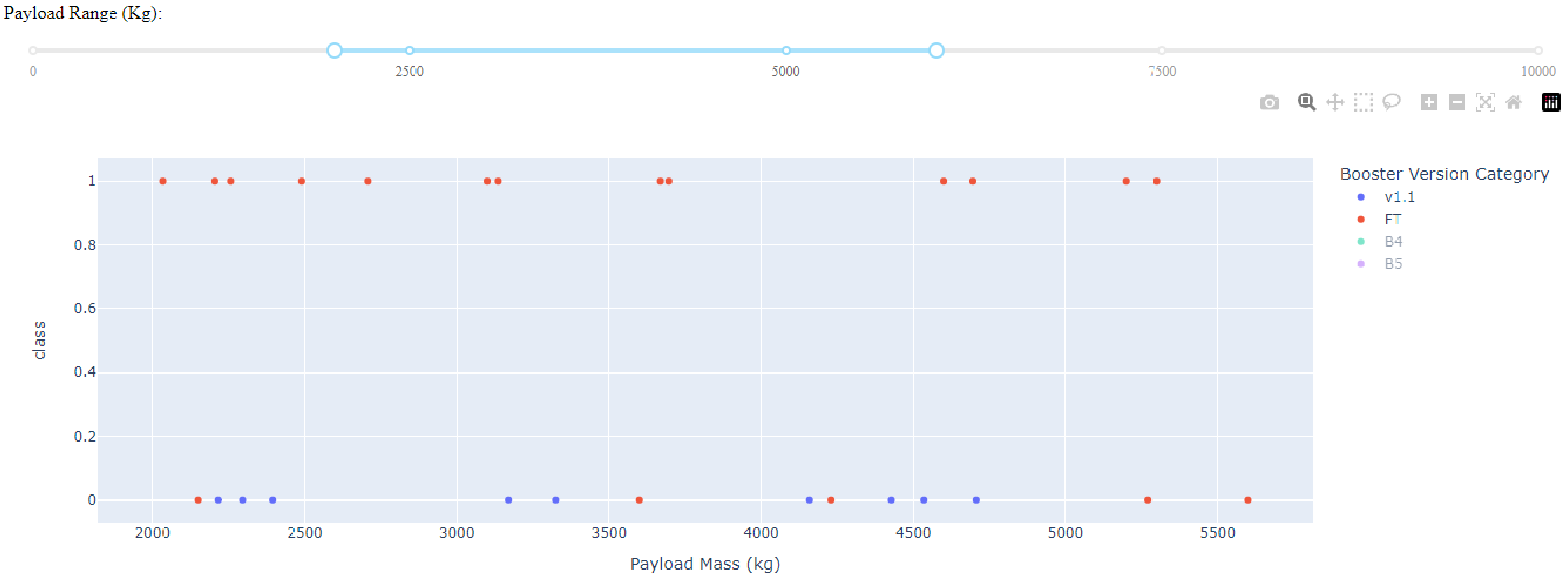


CCAFS SLC-40

Total Success Launches for CCAFS SLC-40

- With the record of being the 2nd with most launches, CCAFS SLC-40 is the launch site with better successful launches rate.
- Near followed by VAFB SLC-4E, with 2% less.

SpaceX Launch Records Dashboard:



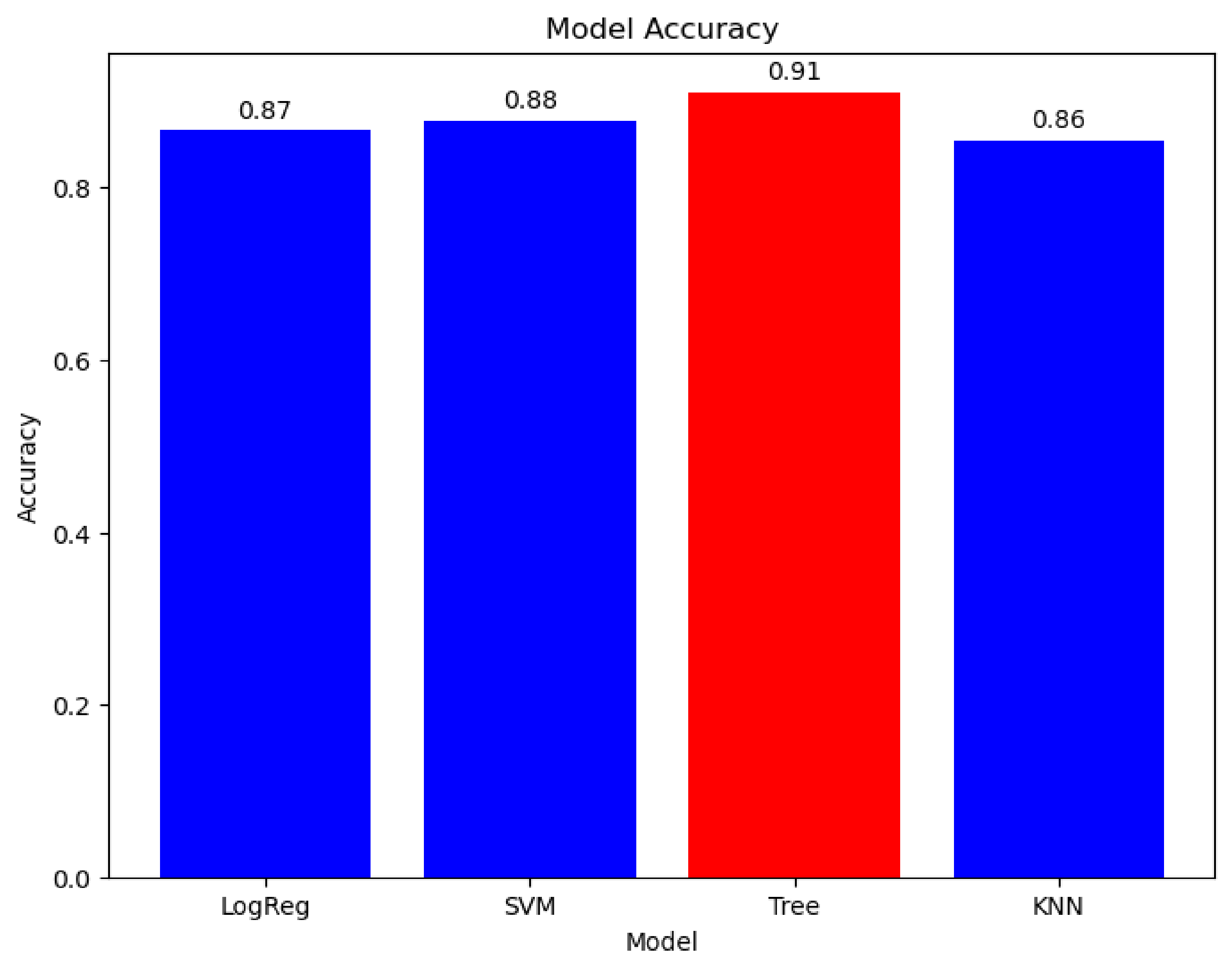
Payload Range (Kg):

- The Payload Mass Range with the most information to infer is between 2000 and 6000 Kg, where:
  - V1.1. is the Booster with the worst success rate.
  - FT is the Booster with the better success rate.

Section 5

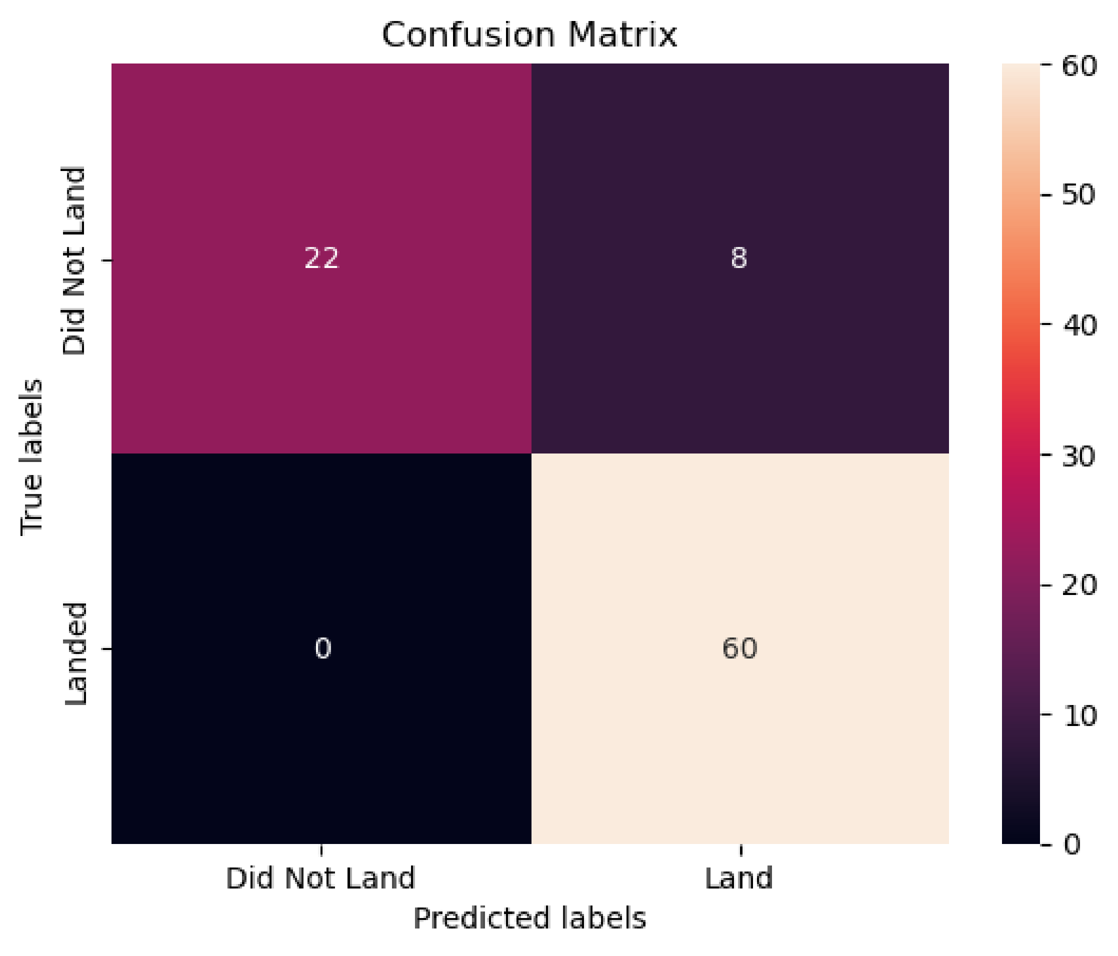
Predictive Analysis (Classification)

Classification Accuracy



- **When fitted with the whole datasets:**
  - Models shows that the Decision Tree has the highest accuracy between all of them.
  - Nevertheless, this was achieved after several iterations since the random nature of the Tree algorithm implies different organizations within features order.
  - Hence, the most consistent results are given by Support Vector Machine model.

Confusion Matrix

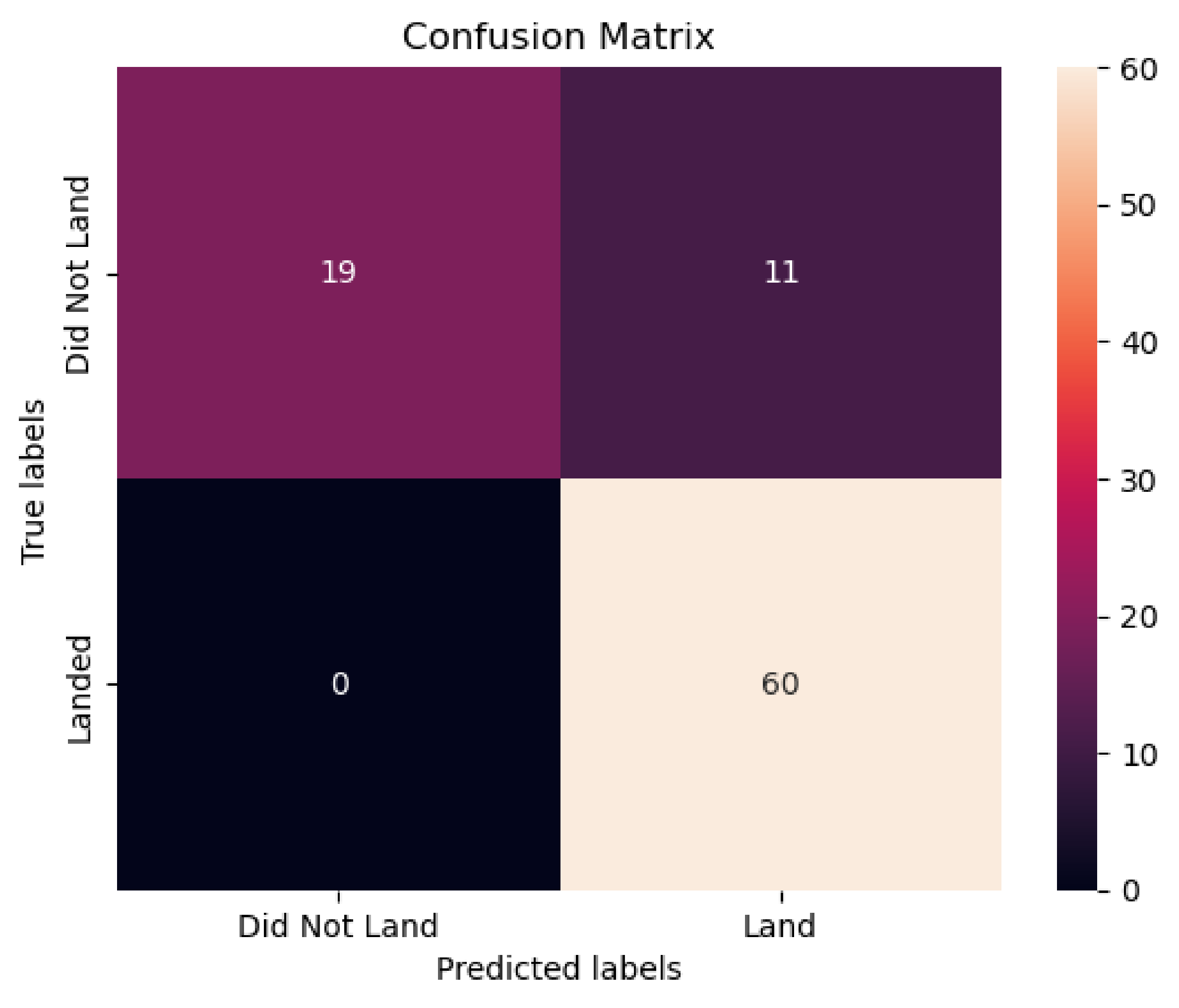


- When deployed with the whole dataset, the confusion matrix shows that its main issue is that it detects False Positives.
- Hence, the model tends to overestimate the success rate of landing outcomes.
- I may have to do with the increasing success rate of the landings with each launch.

Conclusions

- SpaceX has been learning over the time to effectively control the landing outcomes.
- Nevertheless, it's exactly for that that we can predict their success landing rate by looking at, for example, the launch site where the next mission are going to occur.
  - And in case that they'll try it in a new launch site, an excellent determinant of the successful rate would be if the place is either or not near important infrastructures.
- The best performing models for predicting a future landing outcome are , in descending order:
  1. Decision Tree: Higher Precision (Peak > 90%)
  2. Support Vector Machine: Best Consistency (~88%)

Appendix



- The last point in the Confusion Matrix Analysis is validated by the fact that the same predictive issue is shown in the SVM matrix from the left

Thank you!